

Iniciació a l'administració de sistemes

Dept. Matemàtica Aplicada i Anàlisi,
Universitat de Barcelona

© Sergio Escalera, David Masip

Índex del curs

| | |
|--|-----|
| L'Administrador | 3 |
| El procés d'instal·lació i arrancada | 19 |
| Gestió d'usuaris | 34 |
| Gestió de recursos | 98 |
| Bash 1 | 153 |
| Bash 2 | 196 |
| AWK | 222 |
| Python | 257 |
| Python per administrar | 293 |
| Monitorització | 306 |
| Serveis de Xarxa 1 | 378 |
| Serveis de Xarxa 2 | 424 |
| Firewall | 486 |

L'Administrador

Administració de Sistemes Operatius
Dept. Matemàtica Aplicada i Anàlisi,
Universitat de Barcelona

Contingut

- L'administrador
- Tasques de l'administrador
- Paper de l'administrador dins de l'organització

L'administrador

- Persona responsable del funcionament i manteniment del sistema informàtic
- Responsable d'establir les polítiques, accions i normes per aconseguir que el sistema informàtic sigui:
 - Fiable
 - Segur
 - Eficient
 - De fàcil ús

Capacitats de l'administrador

- Expert manipulant el sistema que administra
- Ha d'ésser capaç d'interpretar les directrius que li comuniquen els superiors i les necessitats dels usuaris
- Capaç d'anticipar-se als problemes (proactivitat)
- Persona organitzada i responsable

Habilitats de l'administrador

- **Habilitats i coneixements**
 - Coneixement del SO
 - Coneixement de les comandes
 - Configurar nous dispositius
 - Analitzar el codi font
 - Crear un nucli a partir del codi font
 - Conèixer la shell
- **Servei a l'usuari**
 - Atendre les necessitats dels usuaris
 - Mantenir un diari amb les interaccions en el sistema
 - Escriure els canvis, problemes, solucions
- **Capacitat per resoldre problemes i canvis freqüents**
 - Treballar en un entorn amb constants interrupcions per part dels usuaris
 - Treballar amb un entorn de constants canvis
 - Canviar de una tasca a una altra durant tot el dia

Tasques de l'administrador

- Administració de comptes d'usuaris
- Manteniment del Hardware i el Software
- Administració, planificació i gestió de la xarxa

Disciplina de treball

- **No improvisar**
 - Tot el que es fa ha d'ésser planificat amb anterioritat
- **Accions reversibles**
 - S'ha de disposar de procediments per poder tornar enrera
- **Accions incrementals**
 - Els canvis han d'ésser de manera incremental i sense produir traumes
- **Primer fer un assaig i després actuar**
 - Provar, provar i provar abans de fer qualsevol canvi o acció en el sistema
- **Tenir coneixements del que es va a fer**
 - Saber com funcionen els canvis i les accions a realitzar

Disciplina de treball

- **Automatitzar les tasques**
 - Conèixer espai en disc, realitzar còpies, comportament del sistema, manteniment de les comptes d'usuari, funcions rutinàries
- **Documentar tot**
 - Polítiques, procediments, canvis, etc.
- **Informar quan sigui possible**
 - Que es va a fer, que s'està fent i que es va fer

Paper de l'administrador dins de l'organització

- **L'organització es compon de:**
 - **Recursos humans:** persones (p.e., usuaris, personal en cap, equip de treball)
 - **Recursos materials:** sistema informàtic (p.e., equips informàtics)
 - **Informació:** Gestió i manteniment de la informació (p.e., arxius informàtics)
 - **Coneixement:** Sistema informàtic i polítiques d'empresa (p.e., regles de negoci)
- **L'administrador forma part d'una organització, no treballa aïlladament**

Interlocutors de l'administrador

- **Els usuaris**
 - Són els que utilitzen el sistema informàtic manegat per l'administrador
- **Els superiors en l'organització**
 - Poder ser: el cap de departament d'informàtica, el gerent, el director tècnic, etc.
 - Dicten les polítiques d'ús dels recursos, des del punt de vista de l'organització

Principis d'actuació

- Responsabilitat
- Autoritat
- Obediència als superiors
- Esperit de servei cap als usuaris

La responsabilitat

- **Reconèixer la responsabilitat**
 - L'administrador ha d'assumir que controla uns recursos crítics per al funcionament de l'organització
 - També que manega informació que pot ser privilegiada, confidencial o sensible
- **Per tant:**
 - No defugir quan hi ha problemes en el seu àmbit de responsabilitat
 - No actuar de forma imprudent
 - No violar les normes establertes per la llei

La autoritat

- **L'administrador és un expert que sap com aconseguir que el sistema funcioni amb un rendiment òptim**
 - El usuaris han de reconèixer aquesta autoritat
 - L'administrador ha de fer valer l'autoritat quan sigui necessari
 - L'autoritat no s'ha de confondre amb les males maneres

Obediència

- La gerència o la direcció tècnica coneixen l'organització en la seva amplitud i defineixen polítiques d'ús del sistema que van més enllà de la part tècnica.
 - No inventar polítiques d'ús de recursos que contradiguin el que diu gerència
 - L'administrador no és el "rei" del sistema
 - L'obediència no ha de ser cega: si es considera que una política no es pot implantar o es pot millorar, sempre s'ha de dialogar amb els superiors

Esperit de servei

- **L'administrador és un instrument per a millorar la vida dels usuaris**
- **L'administrador ha de:**
 - Ajudar en els problemes que tinguin els usuaris
 - Atendre les queixes amb educació
 - No oblidar que els usuaris no són experts informàtics, i per tant cometem errors i no entenen el sistema tan fàcilment com nosaltres

Comunicació amb els usuaris

- Cal que sigui continua, fluida i bidireccional
- Correu electrònic
- Diàleg interactiu (write, read and talk)
- Enviament de missatges (wall)
- Comunicacions a l'inici de la sessió (/etc/motd)

El procés d'instal·lació i arrencada

Administració de Sistemes Operatius
Dept. Matemàtica Aplicada i Anàlisi,
Universitat de Barcelona

Index

- Instal·lació del sistema Operatiu
- Gestor d'arrencada
- Instal·lació d'aplicacions
- Estructura de directoris

Procés d'instal·lació

- Existeixen diferents maneres d'instal·lar el S.O. Aquí seguirem la instal·lació seguint menú en mode text.
 - Mitjà d'instal·lació
 - Tipus d'instal·lació
- Normalment es útil instal·lar només la base i després anar afegint SW a mesura que es fa necessari.
- Configuració inicial mínima
 - Tot el procés de configuració es pot fer un cop el sistema base esta instal·lat.

Procés d'instal·lació: Passos

1. Elecció del mitjà d'instal·lació
2. Configuració bàsica: teclat
3. Particionat del disc dur: es perdran **TOTES** les dades del disc
4. Instal·lar el nucli del sistema i els mòduls
5. Configuració dels dispositius
6. Configurar la xarxa
7. Instal·lació dels paquets SW
8. Configurar els paquets que ho requereixin
9. Configurar el gestor d'arrencada
10. Establir el password de root
11. Crear una conta d'usuari

Procés d'instal·lació: Passos

1. Elecció del mitjà d'instal·lació.

- Cal carregar el programa dbootstrap que fa la configuració inicial del sistema. Es pot fer desde CDROM, floppy o un cargador de linux
- Instal·lació del sistema base

Procés d'instal·lació: Passos

3. Particionat del disc dur

- Establir el sistema de particions del disc i assignar-ne l'espai reservat.
- Unitats en Linux
 - Disquetera: `/dev/fdX`, $X \in \{0, \dots, N \text{ disqueteres}\}$
 - Disc dur IDE: `/dev/hdYX`, $Y \in \{a, b, c, d\}$, $X \in \{1, \dots, N \text{ particions}\}$.
Ex: `/dev/hda1`, és la primera partició del disc dur primari
 - Disc dur SCSI: `/dev/sdYX`

Procés d'instal·lació: Passos

3. Particionat del disc dur

- Particions usuals
 - Depèn de la finalitat de la màquina
 - Root /
 - /home: Arxius dels usuaris
 - /usr : Aplicacions de la distribució
 - /tmp : arxius temporals --> noexec
 - /var : informació log variable
 - /swap: Àrea d'intercanvi --> paginació
 - Recomanació: el doble de la memòria RAM
- Formatar les particions. **¿qué significa?**

Procés d'instal·lació: Passos

- Formatar les particions. ¿qué significa?
 - FAT
 - NTFS
 - Ext2
 - Ext3 --> journaling
 - fsck va correccions per l'estat dels logs
 - Ext4 (nou!!)
 - Més longitud de paraula
 - Compatibilitat alta
 - Més profunditat de directoris
 - Journal millorat
 - etc

Procés d'instal·lació: Passos

4. Instal·lar el nucli (linux) del sistema i els mòduls
5. Configuració dels dispositius --> detecció HW
6. Configuració de la xarxa
 - i. Dispositiu de xarxa eth0
 - ii. Adreça IP
 - iii. Mascara de red
 - iv. Adreça Gateway
 - v. Adreça DNS--> DHCP
7. Instal·lació dels paquets SW
8. Configuració dels paquets

Procés d'instal·lació: Passos

9. Establir el password de root

10. Crear un compte d'usuari

i. Normalment es treballa amb un usuari NO root

ii. Implicacions de seguretat

i. passwd

11. Instal·lació de la resta del sistema.

12. Reboot

Gestor d'arrencada

- **Bootstrapping:** Procés que va desde que engeguem la màquina fins que està llesta per utilitzar
- Passos:
 - Detecció del HW i execució del firmware del PC
 - Localització i execució del programa de boot
 - Localització i execució del kernel del SO
 - El kernel s'inicialitza i prepara el hardware (dispositius, drivers, etc...)
 - El kernel executa el proces **init**

Gestor d'arrencada

- El kernel executa el proces **init**:
 - El init engega els processos de sistema, dimonis, etc.:
 - **init** carrega **inittab**
 - **inittab** carrega **/etc/rcN.d** (run level)
 - » 0 apagar
 - » 6 reiniciar
 - » 1 ó S → monusuari/root
 - S'executen els serveis del run level: **/etc/init.d**
 - » **Alternativa Ipstart**
- S'engega un terminal
- Usuari
- Es carrega perfil d'usuari (ex. bash -> profile)

Gestor d'arrencada

- Boot: Grub/lilo
 - Localitzat en una regió concreta del disc
 - Carrega el kernel de linux
 - GRUP → no necessita modificar MBR amb canvis
- Missatges d'arrencada
 - Comanda dmesg
 - Imprimeix els missatjes del buffer del kernel

Instal·lació d'aplicacions

- Comandes
 - Apt-get
 - etc/apt/sources.list
 - dpkg
- Diferents gestors de paquets
 - tasksel
 - aptitude
 - Synaptics, etc.
 - Utilitzen comandes apt-get i dpkg per la descàrrega i instal·lació de paquets

Estructura de directoris

- Estructura bàsica de directoris

- El directori arrel

- bin Binari de comandes essencials
 - boot Arxius estàtics del carregador d'arrancada(boot-loader)
 - dev Arxius de dispositius
 - etc Configuració del sistema local/màquina
 - home Directoris home dels usuaris
 - lib Llibreries compartides
 - mnt Punt de montatge de particions temporals
 - root Directori arrel de l'usuari root
 - sbin Binari del sistema essencials
 - tmp Arxius temporals
 - usr Segona jerarquia major
 - var Informació variable

Gestió d'usuariis

Administració de Sistemes Operatius
Dept. Matemàtica Aplicada i Anàlisi,
Universitat de Barcelona

Index

- Introducció
- Característiques d'un compte
- Usuaris i Grups
 - Fitxers associats
- Creació i eliminació d'usuaris
- Permisos / Llistes de control d'accés
- Eines automàtiques d'administració de comptes d'usuari
- Gestió de passwords
- Configuració de l'entorn
- Autenticació usant PAM
- Serveis de directori: LDAP
- Network Information Service (NIS)
- Resum de comandes

Introducció

- L'administració de comptes d'usuari és una de les tasques més importants.
- Un compte és la manera com els usuaris es presenten al sistema, accedeixen als seus recursos, i demostren que són qui diuen ser.
- Un **grup** és un conjunt d'usuaris que comparteixen unes característiques comuns.

Introducció

- Un **usuari** d'un sistema (desde el punt de vista administratiu) és una entitat que pot executar programes i/o posseir fitxers.
 - No necessàriament és una persona (**usuari real**), pot ser un usuari especial (**usuari lògic**) associat a algun servei del sistema (**pseudo-usuaris**).
- **Root**: Superusuari. Usuari especial que té els privilegis d'administració del sistema.

Característiques d'un compte

- Té un nom i un identificador d'usuari (UID) únics en el sistema
- Pertany a un grup principal
- Pot pertànyer a altres grups d'usuaris
- Pot definir-se una informació associada a la persona propietària del compte
- Té associat un directori personal per a les dades de l'usuari
- L'usuari utilitza un intèrpret de comandes a la seva connexió, on executarà les aplicacions i les utilitats del sistema operatiu
- Ha de tenir una clau d'accés personal difícil d'identificar
- Té un perfil d'entrada propi, on es defineixen les característiques inicials del seu entorn
- Pot tenir una data de caducitat
- Poden definir-se quotes de disc per cada sistema d'arxius

Usuaris i Grups

- Quan es crea un nou usuari cal:
 - Assignar-li un identificador únic d'usuari **UID**.
 - Assignar l'usuari a un o més grups (amb el seu identificador de grup corresponent **GID**).
- Sistema de permisos:
 - Un fitxer o directori té associats uns permisos per al seu propietari, pel grup, i pels altres usuaris:

```
ls -l /usr/bin/amarok
```

```
-rwxr-xr-x 1 root root 31140 2005-03-29 06:27 /usr/bin/amarok
```

Indica que el fitxer correspon a root, al grup root, i els seus permisos de lectura, escriptura i execució.

Usuaris i Grups

- Fitxer implicats a la creació d'usuaris:
 - /etc/passwd: Es guarda la informació de les comptes d'usuari.
 - /etc/shadow: Guarda la informació dels passwords (encriptats).
 - /etc/groups: Conté les definicions de grups
 - /etc/gshadow: Manté la informació dels passwords dels grups.

Usuaris i Grups

/etc/passwd

- Conté la informació bàsica de tots els usuaris de la forma:
`username:pass:UID:GID:info:home-dir:login-shell`
 - Username: Identificador/nom únic i públic del usuari.
 - Pass: Password del usuari encriptat. Si s'utilitza un fitxer shadow, aquest camp es substitueix per una `x`.
 - UID: Número identificador de l'usuari.
 - GID: Número primari de grup assignat al usuari
 - Info: Camps d'informació del usuari, nom, telefon, ... Permet guardar fins a 5 camps separats per coma.

Usuaris i Grups

`/etc/passwd`

- `home-dir`: Directori personal de l'usuari on guardarà els seus fitxers, on iniciarà les sessions.
- `login-shell`: L'interpret de comandes que s'executarà per l'usuari quan entri a la màquina. Ex: `/bin/sh` (Bourne shell), `/bin/csh` (C shell), `/bin/ksh` (Korn shell).
 - A `/etc/shells` hi ha una llista de tots els disponibles

Usuaris i Grups

/etc/shadow

- Base de dades que guarda el password encriptat corresponent a cada compte d'usuari.
 - El /etc/passwd acostuma a ser public i “llegible”, usuaris maliciosos podrien “crackejar” passwords dèbils.
 - El /etc/shadow només és accessible pel root.

username:pass encrypt:informacio-dades

- On hi ha el id d'usuari, el password encriptat (DES, MD5,...), i informació de quan s'ha posat el password:
 - Data de l'últim canvi, dies per canviar de clau, dies per avisar a l'usuari per canvi de clau, etc.
 - i la seva caducitat.

Usuaris i Grups

- Grups: Permeten a un conjunt d'usuaris compartir fitxers, i recursos en general.

/etc/group

- Conté la informació associada a un grup:

`nom:x:GID:usuaris`

- Nom: un identificador del grup. Exemple: docents, research,...
- El x indica que el password està guardat a `/etc/gshadow`.
- GID: el número identificador de grup.
- Llista dels usuaris del grup.

Usuaris i Grups

- Un usuari té com a grup per defecte:
 - El seu grup primari quan crea/escriu un fitxer
 - Qualsevol grup on apareix el seu nom a /etc/group quan llegeix un fitxer.
- Comandes:
 - `groups [usuari]`
 - Dona els grups del usuari actual.
 - `newgrp grup`
 - Canvia el grup del usuari actual.

Usuaris i Grups

- Ús dels grups: Permeten la compartició de fitxers entre usuaris que tenen una certa relació.
 - Organització
 - Grup de treball...
- Problema: Usuari que NOMÉS necessita accedir a un fitxer compartit. No és segur donar-li privilegis per accedir a tot el grup.
 - ACL: Llistes de control d'accés, particular per cada recurs

Usuaris i Grups

/etc/gshadow

Groupname:pass encrypt:group_admin:usuaris_addic.

- GroupName i pass son el nom del grup i el password encriptat
 - group_admin: és la llista d'usuaris que poden modificar el grup
 - Usuaris addicionals.
- Tots els fitxers de grup i usuaris han de tenir el mínim de privilegis possibles

-rw-r--r-- 1 root root 651 2006-03-21 16:49 /etc/group

-rw-r----- 1 root shadow 545 2006-03-21 16:49 /etc/gshadow

-rw-r--r-- 1 root root 1064 2006-03-14 12:09 /etc/passwd

-rw-r----- 1 root shadow 740 2006-03-14 12:09 /etc/shadow

Usuaris i Grups

Comptes d'usuari predefinides en UNIX

- Usuaris standard de sistema, per gestionar diferents serveis: mail, sys, mysql, ...
 - Password deshabilitat (*).
- Grups estàndar definits en UNIX: root, system, wheel, bin, disks, adm, ...
 - 0 0 para root
 - UID
 - 0-99 reservats pel sistema
 - 100-60000 assignats a l'usuari
 - GID
 - 0-99 reservats pel sistema
 - 100-60000 assignats a l'usuari

Usuaris i Grups

Comptes d'usuari predefinides en UNIX

| Usuari | UID | GID | Descripció |
|--------|-----|-----|--|
| root | 0 | 0 | Administrador de control total |
| bin | 1 | 1 | Propietari de les utilitats del sistema operatiu |
| daemon | 2 | 2 | Gestor de serveis generals |
| adm | 3 | 4 | Propietari dels arxius de requisits històrics i administratius |
| lp | 4 | 7 | Administrador dels serveis d'impressió |
| nobody | 99 | 99 | Gestor de serveis varis |
| ftp | 14 | 50 | Controlador de l'accés a l'arbre del servei FTP anònim |
| ssh | 74 | 74 | Usuari fictici gestor del servei SSH |
| apache | 48 | 48 | Propietari dels fitxers i directoris del servei de hipertexte Apache |
| squid | 23 | 23 | Controlador del servei de representació Squid |
| ldap | 55 | 55 | Permet l'accés al servei de directoris LDAP |

Usuaris i Grups

Comptes d'usuari predefinides en UNIX

| Grup | GID | Descripció |
|--------|-----|--|
| root | 0 | Administradors amb control total |
| bin | 1 | Binaris del sistema |
| daemon | 2 | Serveis generals |
| sys | 3 | Control del sistema |
| adm | 4 | Fitxers històrics i administratius |
| tty | 5 | Accés a la consola |
| lp | 7 | Servei d'impressió |
| kmem | 9 | Control de memòria del nucli del sistema |
| man | 15 | Pàgines de manuals |
| ftp | 50 | Servei FTP anònim |
| nobody | 99 | Control de serveis |
| users | 100 | Usuaris normals |
| floppy | 19 | Accés a disquettes |
| apache | 48 | Servei d'hipertexte HTTP |
| squid | 23 | Servei representant |
| ldap | 55 | Servei LDAP |

- Usuaris ficticis han de tenir el seu compte deshabilitat per evitar infiltracions:
 - Bloquejar la clau i assignar `/sbin/nologin` com intèrpret de comandes del compte

Creació d'usuari

- Creació manual d'usuari
 - Conceptes associats
 - Fitxers implicats
- Eines automàtiques de creació d'usuari.

Creació d'usuaris manual

- Tasques a realitzar per l'administrador:
 - Assignar un nom d'usuari, ID, grup primari i altres grups.
Editar els fitxers de configuració: /etc/passwd, /etc/groups,...
 - Assignar un nou password al compte.
 - Crear una carpeta home per l'usuari.
 - Crear els fitxer d'inicialització necessaris.
 - Canviar els permisos del home.
 - Ajustar els paràmetres del compte (caducitat,...)
 - Afegir l'usuari als serveis de l'empresa (impresió, disk,...)
 - Verificació final del compte d'usuari.

Creació d'usuaris manual

- Definir el compte d'usuari:
 - Assignar el id d'usuari (política d'administració).
 - Assignar el grup primari i els altres grups als que pertany.
 - Home directory
 - Login shell (últim camp de /etc/passwd)
 - /bin/sh
 - /bin/csh
 - /bin/bash
 - /bin/false: Provoca la desconnexió de l'usuari
 - Deshabilita un compte.

Creació d'usuaris manual

- **Comptes captius:** S'utilitzen per limitar les accions d'un determinat usuari.
 - Usar un únic programa com a login shell.
 - Usar "restricted shells": Comptes on l'administrador defineix que pot fer l'usuari.
 - Ex: /usr/bin/rsh, rksh, bash -restricted, ...
 - Prohibir l'ús de: cd, set paths, redireccions > >>, accés a rutes amb /.

Creació d'usuaris manual

- Assignar el password
 - `passwd nom-usuari`
 - 8-128 caràcters
 - Consideracions de seguretat al assignar un password.

Creació d'usuaris manual

- Crear el home directory:
 - Afegir la entrada a `/etc/passwd`.
 - Crear la carpeta `mkdir /home/nom-usuari`
 - Assignar els permisos corresponents:
`chown usuari.grup /home/nom-usuari`
`chmod 755 /home/nom-usuari`

Creació d'usuaris manual

- Crear els fitxers d'inicialització:
 - Fitxers que s'executen després del login *o quan es crea un shell*, inicialitzant la shell i l'entorn gràfic.
 - `.profile`, `.login`, `.cshrc`, `.bashrc`, ...
 - Localitzats al home-dir de cada usuari.
 - L'administrador en crea uns per defecte a `/etc/skel`
 - Només cal copiar els per defecte al nou user-home, i ajustar-los a les necessitats.

Creació d'usuaris manual

- Tasques dels fitxers d'inicialització de connexió (login):
 - Establir path. `env PATH $PATH:/usr/local/bin:~/bin`
 - Establir la màscara de creació de fitxers. `umask 022`
 - Establir el tipus de terminal. `env term vt100`
 - Generar les variables d'entorn del usuari.
- Tasques dels fitxers d'inicialització d'una shell (`.cshrc`, `.bashrc`).
 - Establir variables d'entorn.
 - Establir alias de comandes. Ex: `alias ll ls -l`

Creació d'usuaris manual

- Tasques dels fitxers d'inicialització d'entorn gràfic
 - Fitxers genèrics: `.xinitrc`, `.xsession`, `.Xauthority`
 - Inicialitzen la interfície de finestres
- Tasques d'inicialització generals `/etc/profile`
 - S'executa abans del fitxer d'inicialització de login de cada usuari, i es comú per a tots.

Creació d'usuaris manual

- Ajustar els paràmetres del compte:

- Paràmetres associats a la longitud i temps de vida dels passwords, freqüència de login, etc...
- LINUX: S'especifiquen a `/etc/login.defs`. Exemple:

```
ENV_PATH path    #el path per defecte per root i els usuaris
LOGIN_RETRIES 10
LOGIN_TIMEOUT 60
UID_MIN 100
UID_MAX 20000
...
```

Creació d'usuaris manual

- Verificació del compte:

- Intentar entrar com al nou usuari.

- Ús de la comanda `su`

- `su usuari`

Fa el canvi d'usuari conservant els valors anteriors d'entorn, directori actual, etc...

- `su - usuari`

Simula un login complet com a usuari.

Eliminació d'usuari

- Eliminar un usuari.
 - Deshabilitar el compte:
 - Canviar el password per un * al /etc/shadow
 - Comanda: `passwd -l usuari`
 - Eliminar completament el compte de l'usuari:
 - Exigir el canvi de tots els passwords que l'usuari coneixia
 - Finalitzar els processos de l'usuari
 - Eliminar l'usuari dels grups secundaris
 - Eliminar l'arxiu de mail de l'usuari (back up)
 - Definir un alias de mail per l'usuari (nou destí)
 - Eliminar l'usuari de les mailing lists.

Eliminació d'usuari

- Eliminar les tasques programades de l'usuari.
- Eliminar el home-dir de l'usuari (Back up)
- Buscar, gravar i eliminar altres fitxers propietaris.
- Eliminar la quota de disk de l'usuari.
- Eliminar tots els serveis per l'usuari.

Permisos

- Els fitxers tenen associada la informació de pertinença a un usuari i a un grup

- Quan un usuari crea un fitxer és el seu propietari.

```
>ls -l fixter
```

```
-rwxr-xr-- 1 usuari grup 2100 May 13 07:32 fixter
```

- Si vol canviar el propietari o grup:

```
>chown nou_usuari fixter
```

```
>chgrp nou_grup fixter
```

```
>chown nou_usuari:nou_grup fixter
```


Permisos

- Els fitxers tenen associada la informació de protecció d'escriptura, lectura i execució, que són diferents pel **propietari, grup i altra gent**:

```
-rwxr-xr-- 1 usuari grup 2100 May 13 07:32 fitxer
```

- Si vol canviar els permisos d'un fitxer:

```
>chmod u+w fitxer (dona permís d'escriptura a l'usuari)
```

```
>chmod 744 fitxer (dona globalment els permisos rwxr--r--  
en notació octal)
```

- Directoris: execució (poder fer cd directori), lectura (poder fer un ls al directori), escriptura (poder-hi escriure fitxers).

Permisos

- Fitxers de nova creació: permisos per defecte.

– S'utilitza la comanda:

```
umask num_octal      (va al inrevés que chmod 0: activat)
```

```
umask 023  (equivalent a: rwxr-xr--)
```

Xrwxr-xr--

X=

- fitxer

d directori

l enllaç

p canonada

Permisos

- Fitxers:
 - r lectura, l'usuari pot llegir el fitxer
 - w escriptura, l'usuari pot escriure al fitxer
 - x execució, l'usuari pot executar el fitxer
- Directoris:
 - r lectura, l'usuari pot llegir el contingut del directori
 - w escriptura, l'usuari pot crear, modificar i esborrar les entrades del directori
 - x accés, l'usuari pot accedir al directori (cd)

| Comanda | Permisos directori origen | Permisos fitxer | Permisos directori destí |
|---------|---------------------------|-----------------|--------------------------|
| cd | X | No aplicable | No aplicable |
| ls | R | No aplicable | No aplicable |
| mkdir | W,X | No aplicable | No aplicable |
| rmdir | W,X | No aplicable | No aplicable |
| cat | X | R | No aplicable |
| rm | W,X | - | No aplicable |
| cp | X | R | W,X |
| mv | W,X | - | W,X |

Permisos

- Bits specials:
- Fitxers:
 - Setuid: (**SUID** usuari, **SGID** grup) Permet que un fitxer sigui executat segons els permisos del seu creador/grup, no pas segons els permisos del usuari que l'executa.

```
>chmod u+s fitxer (s per propietari)
```
- Directoris:
 - t per al resta d'usuaris: permet que en directoris compartits els fitxers només puguin ser modificats per el propietari (com per exemple en directoris temporals com `\tmp`)
 - s per el grup: els fitxers que es creen en el directori tindran el mateix grup que el propietari del directori, en lloc del grup del propietari

Permisos

- Notació

- La notació simbòlica s'usa per afegir (+), treure (-) ó assignar (=) permisos agrupats segons el tipus.
- La notació numèrica en format octal serveix per assignar tots els permisos a la vegada mitjançant una operació lògica.

| Permisos normals | Valor octal | Notació simbòlica |
|-----------------------|-------------|-------------------|
| Propietari: | | |
| Lectura | 400 | u+r |
| Escriptura | 200 | u+w |
| Execució/accés | 100 | u+x |
| Grup: | | |
| Lectura | 40 | g+r |
| Escriptura | 20 | g+w |
| Execució/accés | 10 | g+x |
| Altres: | | |
| Lectura | 4 | o+r |
| Escriptura | 2 | o+w |
| Execució/accés | 1 | o+x |
| Permisos especials | Valor octal | Notació simbòlica |
| Propietari: | | |
| Usuari actiu (SUID) | 4000 | u+s |
| Grup: | | |
| Grup actiu (SGID) | 2000 | g+s |
| Altres: | | |
| Director d'intercanvi | 1000 | +t |

Permisos

- Llistes de control d'accés

| Tipus | Format | Descripció |
|---------------|------------------------------|---|
| ACL_USER_OBJ | user ::permisos | Drets d'accés per el propietari |
| ACL_USER | user :usuari:permisos | Permisos per un determinat usuari |
| ACL_GROUP_OBJ | group ::permisos | Accessos permisos al grup de l'objecte |
| ACL_GROUP | group :grup:permisos | Permisos per un determinat grup |
| ACL_MASK | mask ::permisos | Indica màscara de permisos efectius que s'aplicaran a les entrades de tipus ACL_USER, ACL_GROUP_OBJ i ACL_GROUP |
| ACL_OTHER | other ::permisos | Drets d'accés per a processos que no compleixen els requisits anteriors |

- setfacl crea, modifica o treu una definició de la ACL.
- getfacl Mostra la definició de llistes ACL
- chacl Modifica definició
 - >chacl u::rw-,g::r--,o::---,u:usuari:r--,m::rw- arxiu

Eines automàtiques d'administració de comptes d'usuari

- Eines de comandes o gràfiques que permeten afegir o eliminar usuaris.
- Línia de comandes:
 - Afegir un usuari

```
useradd -g grup-prim -G grup1, grup2 -s /bin/bash -c "Juan Perez" -m juanp
```
 - Modificar el nom d'un compte:

```
usermod -m -l juanp perep
```
 - Eliminar un usuari:

```
userdel juanp
```
 - Afegir un grup:

```
groupadd jefes
```
 - Canviar de nom un grup:

```
groupmod -n jefes caps
```
 - Canviar els membres d'un grup:

```
gpasswd -a juanp jefes
```

Eines automàtiques d'administració de comptes d'usuari

- Interfícies gràfiques:
 - Linuxconf
 - Ximian Setup tools
 - KDE User Manager
 - Red Hat user Manager
 - etc.

Gestió de passwords

- Tots els usuaris han de tenir un password **SEGUR**
 - Molt fàcil de recordar + molt difícil d'endevinar
 - Passwords automàtics: segurs però difícils.
 - Passwords autogenerats: fàcils però sovint no segurs.
 - Solució: **EDUCAR** als usuaris per generar passwords segurs.
- Què **NO** s'ha de fer:
 - Usar parts/noms del entorn familiar/amics
 - Usar dades públiques (data naixement, matricula...)
 - Dades d'algun esdeveniment important, hobby, pel·lícula,...
 - Dades de l'empresa

Gestió de passwords

- Atacs de *password-cracking*: proven totes les paraules/permutacions d'un diccionari:
 - Evitar usar paraules que apareixen al diccionari
 - Llistes públiques de paraules en diccionaris per crackers
 - Paraules truncades: Ex: diaposit (diapositiva)
 - Noms de gent famosa
 - Passwords públics
 - Qualsevol de les anteriors afegint un número o reordenació.
Ex. miquel, leuqim, miquel1, ...

Gestió de passwords

- Bones pràctiques (no suficients):
 - Afegir al password varis caràcters extra, a ser possible símbols: ,_/(\$:...
 - Crear passwords amb errors intencionats: varsalona
 - Usar majúscules i minúscules (augmenta el número de proves a fer per un cracker automàtic)
 - Concatenar parts de paraules
 - Afegir paraules curtes al mig

Gestió de passwords

- Consells per a la creació de bons passwords:
 - Difícils d'endevinar i fàcils de recordar:
 - Agafar la primera lletra d'una frase fàcil de recordar per a l'usuari:
“Tinc un hipopòtam a la banyera de casa”
Password: Tuhalbdc o millor T1halBdC
 - Crear passwords usant un desplaçament físic del teclat.
Holabondia -> jpñsnpmfos
(movent només una tecla a la dreta)

Gestió de passwords

- No pot haver-hi comptes d'usuari sense password
- Establir una mida mínima del password (≥ 8)
- Canviar de password si:
 - Hi ha sospites que algú més el sap.
 - Quan un usuari marxa de l'empresa.
 - Quan el root marxa: canviar TOTS els passwords
 - Quan algú ha tingut accés al /etc/shadow
- Canviar periòdicament el password de root.
`chage -d 0 -M 999 #Obliga a tothom a canviar el password`
- Tenir un password diferent per cada compte obert.
 - Usar algun algorisme: Un password complicat de base i afegir un canvi incremental a partir de dades fàcils de recordar.

Gestió de passwords

- Mecanismes del administrador:
 - Establir una mida mínima del password.
 - Normalment integrat en la comanda `passwd`.
 - `/etc/login.defs`, habilitar el flag `OBSCURE_CHECK_ENAB` – additional checks
 - Controlar que l'usuari no entri passwords massa senzills.
 - Utilitat `npasswd` \approx `passwd`, però a més estableix controls amb la resta d'informació de tots els usuaris per evitar coincidències.
 - Ús de llistes de passwords històrics: evitar que l'usuari recicli el password anterior.
 - Bloquejar un compte després d'excessius intents fallits de login.

Gestió de passwords

– Establir caducitat als comptes.

• Per defecte /etc/login.defs

```
PASS_MAX_DAYS    100    #canviar cada 100 dies
PASS_MIN_DAYS    1      #minim un dia
PASS_WARN_AGE    7      #Avisar quan faltin 7 dies
PASS_MIN_LEN     8      #Minim de 8 characters
OBSCURE_CHECKS_ENABLE  yes #filtrat els "facils"
PASS_CHANGE_TRIES      3  #3 Intents
PASS_ALWAYS_WARN      yes #avisar a root
PASS_MAX_LEN          8  #maxima longitud
CRACKLIB_DICTPATH    /usr/lib/diccionari_crack #diccionari
```

• Per a cada usuari en particular: /etc/shadow

```
user:passwd_enc:ultim_canvi:min_vida:max_vida:avis:inactiu:caducitat
```

Gestió de passwords

- Programes per comprovar la debilitat dels passwords:
 - **John the Ripper**
 - Paquet que donat un diccionari i un fitxer de passwords, els intenta endevinar usant un gran nombre de transformacions entre les paraules.
 - Exemple:

```
umask 077
```

```
unshadow /etc/passwd /etc/shadow > mypasswd
```

```
john --wordlist=diccionari.lst --rules mypasswd
```


Gestió de passwords

- Programes per comprovar la debilitat dels passwords:

- **crack**

- Exemple:

```
umask 077
```

```
unshadow /etc/passwd /etc/shadow > mypasswd
```

```
crack -nice 5 mypasswd
```

- Crea un conjunt de diccionaris i intenta trobar els passwords a partir de les paraules i diferents transformacions (configurables) entre elles.

Configuració de l'entorn

- Entorn global:
 - L'administrador ha de crear un perfil d'entrada comú (`/etc/bashrc` en el cas de BASH), on es defineixen les variables del sistema i s'executen fitxers de configuració propis de cada aplicació.
- Perfil personal:
 - És l'últim pas de connexió de l'usuari (`~/.bash_profile` per BASH)

Configuració de l'entorn

| Tipo de Shell | Shell estándar | Clones libres |
|--------------------|----------------|------------------|
| AT&T Bourne shell | sh | ash, bash, bash2 |
| Berkeley "C" shell | csh | tcsh |
| AT&T Korn shell | ksh | pdksh, zsh |
| Otros interpretes | -- | esh, gush, nwsh |

Interpretes de comandos en Linux/Unix

| Shell | Del sistema (globales) | | Privados (personales) | |
|-------|------------------------|----------------|------------------------|-----------------|
| | Siempre | Login shell | Siempre | Login shell |
| csh | /etc/csh.cshrc | /etc/csh.login | ~/.cshrc | ~/.login |
| tcsh | /etc/csh.cshrc | /etc/csh.login | ~/.tcshrc ^a | ~/.login |
| sh | -- | /etc/profile | -- | ~/.profile |
| bash | -- | /etc/profile | ~/.bash_rc | ~/.bash_profile |
| zsh | /etc/zshrc | /etc/zlogin | ~/.zshrc | ~/.zlogin |

a. Si no existe, el tcsh intenta leer el fichero ~/.cshrc.

Es decir, todos los usuarios que ejecuten el entrar (login) el csh o tcsh, ejecutarán el script situado en /etc/csh.login y /etc/csh.cshrc si estos dos ficheros existen. Aparte, cada usuario puede configurar su entorno de trabajo mediante los ficheros personales .tcshrc o .cshrc y .login, ubicados en su directorio HOME personal.

Autenticació Usant PAM

- Pluggable Authentication Modules (PAM): Esquema que permet l'autenticació en entorns UNIX desde múltiples contextos (no només al moment de login), independitzant les aplicacions del sistema d'autenticació subjacent.
 - Les aplicacions criden als mòduls d'autenticació en temps d'execució (es resol mitjançant LDAP, domini Samba o NT, connexió NIS, NIS+, autenticació Kerberos, etc.)
 - Elements:
 - Aplicacions o serveis que usin autenticació
 - Mòduls o llibreries que fan les tasques d'autenticació. /usr/lib/security.
 - Fitxers de configuració per cada servei.
 - Específics: /etc/pam.d/servei (ex: /etc/pam.d/login)
 - Globals: /etc/pam.conf

Autenticació Usant PAM

- Exemple: Configuració de PAM per al servei SU:

```
auth sufficient /lib/security/pam_rootok.so
auth required /lib/security/pam_wheel.so
auth required /lib/security/pam_unix.so shadow nullok
account required /lib/security/pam_unix.so
password required /lib/security/pam_unix.so
session required /lib/security/pam_unix.so
```

Autenticació Usant PAM

- Exemple: Configuració de PAM per al servei SU:
 - Tipus d'entrada:
 - Auth: Especifica procediments d'autenticació.
 - Account: Establir atributs del compte.
 - Password: Canvi de password en el contexte.
 - Session: Usat per a login
 - Segon camp: Com afecten els resultats a l'autenticació:
 - Sufficient: Si és correcte, es salta la resta.
 - Requisite: Si falla, es salta la resta i no autentica.
 - Required: El modul ha de funcionar per a que el procés d'autenticació sigui acceptat.
 - Optional: Usat si no hi ha resposta d'altres mòduls

Autenticació Usant PAM

- Exemple: Configuració de PAM per al servei SU:

```
auth sufficient /lib/security/pam_rootok.so
```

Mira si és root, si ho és retorna OK (sense mirar més: sufficient). El mòdul pam_rootok.so se'n ocupa.

```
auth required /lib/security/pam_wheel.so
```

Si no és root, es mira si és d'un grup permès. Es limita l'ús del su als usuaris del grup del sistema.

```
auth required /lib/security/pam_unix.so shadow nullok
```

Es mira el password introduït per l'usuari. Es imprescindible encertar.

...

Autenticació Usant PAM

- Moduls PAM en Linux:
 - Existeixen multitud de mòduls PAM:
 - pam_permit: Permetre accés
 - pam_pwdb: Verificar/canviar passwords
 - pam_cracklib: Usats per mirar passwords massa febles.
 - pam_env: Manipular variables d'entorn
 - pam_limits:...
 - Més Informació:
<http://www.kernel.org/pub/linux/libs/pam/modules.html>

Servei de Directori (LDAP)

- LDAP: Lightweight Directory Acces Protocol. És un protocol de xarxa que permet accés a un servei de directori ordenat i distribuït on es pot trobar diversa informació sobre la xarxa i els seus usuaris. ≈ Base dades de la xarxa.
 - Està optimitzat per l'accés de lectura.
 - Les estructures de dades que manega són extensibles segons les necessitats locals.
 - Standard RFC 4510 (X.500 directory service)
 - Emmagatzemament distribuït.
 - Accés per TCP/IP

Servei de Directori (LDAP)

- Components:
 - Daemons: slapd (dimoni OpenLDAP) i slurpd (replicació de dades).
 - Base de dades (GNU GDBM o Berkeley BD)
 - Utilitats: Idapadd (afegir entrades) Idapmodify (modificar entrades) Idapdelete, Idapsearch, Idappasswd.
 - Fitxers de configuració: /etc/openldap.
- Directory:
 - Registres (records) → entrades
 - Camps → atributs.

Servei de Directori (LDAP)

- Exemple OpenLDAP directory:

Dn: cn=Juan Perez, ou=Llista, dc=maia, dc=es

ObjectClass: person

Cn: Juan Perez

Sn: Perez

Description: Expert en NTFS LDAP i NIS

TelephoneNumber: 56

- Format de dades LDIF (LDAP data Interchange Format).
 - La primera línia fa de clau única, especificant les dades com a parelles atribut-valor separades per comes.
 - Tipus d'entrada: valor.

Servei de Directori (LDAP)

- Esquemes: Serveixen per definir la estructura de les entrades.

```
include /etc/openldap/schema/core.schema (slapd.conf)
```

- Exemple

```
Objectclass (2.5.6.6 NAME 'person' SUP top STRUCTURAL
MUST (sn $ cn)
MAY (userPassword $ telephoneNumber $ seeAlso $ Description ) )
```

Atributs:

```
attributetype (2.5.4.4 NAME ('sn' 'surname' ) SUP name)
attributetype (2.5.4.41 NAME 'name'
EQUALITY caseIgnoreMatch)
```

Més informació: <http://www.openldap.org/doc/admin23/>

Servei de Directori (LDAP)

- Aplicació: Autenticació d'usuaris.
 - Triar l'esquema adequat.
 - Migrar les dades sobre els comptes actuals al directori
 - Configurar PAM per usar LDAP (mòdul pam_ldap.so).
 - Configurar l'accés a cada recurs (entrades slapd.conf).

Més informació:

<http://www.openldap.org/doc/admin23/slapdconf2.html#Access%20Control>

Network Information Service (NIS)

- Network Information Service (NIS): protocol client servidor que permet mantenir els fitxers de configuració d'una xarxa completa.
 - Dissenyat assumint confiança entre els sistemes de la xarxa: No segur.
 - Manté un directori central d'usuaris, hostnames, i altres paràmetres de configuració.
 - Reemplaçat per LDAP.
- Comandes:
 - yppasswd: per canviar el password contra la BD remota (yp: yellow pages).

Resum de comandes

| Comanda | Descripció |
|----------|---|
| useradd | Nou compte d'usuari |
| usermod | Modifica els paràmetres d'un compte |
| userdel | Esborra un compte d'usuari |
| passwd | Modifica la clau d'accés a un compte |
| chpasswd | Canvia la clau a varis usuaris usant un fitxer d'entrada de dades |
| chage | Canvia les restriccions temporals d'un compte |
| chfn | Canvia la descripció de l'usuari |
| chsh | Canvia el intèrpret de comandes del compte |
| groupadd | Crea un nou grup d'usuaris |
| groupmod | Modifica els paràmetres d'un grup d'usuaris |
| groupdel | Esborra un grup d'usuaris |
| gpasswd | Canvia la clau a un grup privat |

Gestió de Recursos

Administració de Sistemes Operatius
Dept. Matemàtica Aplicada i Anàlisi,
Universitat de Barcelona

Index

- Introducció
- Gestió de discs
- Sistemes de fitxers
- Gestió del espai: cuotes
- Impressores i cues d'impressió
- Entorn gràfic: X

Introducció

- En aquest tema s'introduiran els diferents recursos de que disposa un servidor, així com les eines bàsiques de configuració i control associades.
 - Particionat i sistemes de fitxers.
 - Muntatge de nous dispositius d'emmagatzemament.
 - Espai de disc.
 - Gestió de recursos d'impressió.
 - Sistemes de finestres.

Gestió de discs

- La unitat bàsica d'emmagatzemament en un sistema operatiu es la **partició**.
 - Cada sistema de fitxers es crea en una partició.
 - En Unix, tots els sistemes de fitxers es combinen en una estructura d'arbre de directoris comú.
- Les particions tenen establerta una mida fixada en temps de creació.
- Es possible combinar múltiples discs físics en un disc lògic, que alhora es pot subdividir en particions lògiques.

Gestió de discs

- Inserció de nous discs al sistema:
 - Connectar físicament el disc al sistema (bus + alimentació).
 - Instal·lar el driver pel nou dispositiu. Pot ser necessari recompilar el kernel o carregar un mòdul.
 - Definir el sistema de particions del nou disc.
 - Crear el sistema de fitxers de cada partició.
 - Comprovar la integritat del nou sistema de fitxers: `eina fsck`
 - Inserir el nou sistema de fitxers en el fitxer de configuració del sistema.
 - Muntar el sistema de fitxers.
 - Preparar les tasques d'administració associades:
 - Introduir el nou dispositiu a la política de back up.
 - Preparar els límits de quotes de disc.

Gestió de discs

- Connexió física de dispositius
 - Existeixen dos grans famílies de dispositius de disc:
 - IDE: Fàcils d'instal·lar i configurar (Master/Slave jumper). Mercat d'ordinador de sobretaula.
 - SCSI: Més cars i complexes, pensats pel seu ús en servidors. Rendiment més elevat.

Mes informacio: <http://www25.big.jp/~jam/PC/SCSI-IDE.html>

- Creació del fixter associat al nou dispositiu
 - Sovint ja venen al directori /dev del sistema
 - Crear-los: `mknod nom {c|b} major minor`
o `be MAKEDEV nom` (script més simple)
 - Conèixer el nom del dispositiu: `dmesg` al connectar-lo.
Detected scsi disk **sda** at scsio, id 2, lun 0
`scsi : detected 1 SCSI disk total`

Gestió de discs

- Definició dels sistemes de les particions i sistemes de fitxers: L'administrador ha de planificar amb cura l'espai dedicat a cada partició en funció de les necessitats que ha de cobrir.
 - El rendiment global del sistema pot baixar considerablement si es fa una mala planificació del sistema de fitxers.
 - Depèn dels usos que es donin al sistema per part de la organització.
 - Consell:
 - Parts del sistema usades molt freqüentment (BD, Web,...), haurien d'estar en sistemes de fitxers ubicats en dispositius diferents, endollats a **controladors** diferents.
 - L'àrea de swap no hauria d'estar ubicada en el mateix dispositiu.

Gestió de discs

- Definició i creació de les particions: Utilitats
 - `diskdruid` : Només en el moment d'instal·lar el sistema.
 - `fdisk` i `cfdisk`: Es poden cridar desde línia de comandes.
- Tipus de Particions:
 - Primàries: Només hi poden haver 4 particions primàries al disc. Raons històriques, la taula de particions de disc només té espai per a 4 registres.
 - Exteses: Es consideren com particions primàries, i només hi pot haver una partició extesa a cada disc.
 - Lògiques: La partició extesa es pot subdividir en particions lògiques.

Gestió de discs

- Creació del sistema de fitxers: Un cop definides les particions amb `fdisk`, cal crear cada un dels sistemes de fitxers. Utilitat **mkfs**.
 - **mkfs** -t <tipus> <opcions> <dispositiu>: Crea un sistema de fitxers especificat.
- Utilitats de sistema:
 - `tune2fs`: Utilitat que permet alterar algunes de les propietats del sistema de fitxers.
 - `resize2fs`: Utilitat que permet modificar la mida d'un sistema de fitxers.
 - Augmentar la mida d'un sistema de fitxers sempre és segur.
 - Disminuir-la pot comportar pèrdua de dades

Sistemes de Fitxers: Introducció

- Les tasques de l'administrador referents al sistema de fitxers són:
 - Crear i posar a disposició dels usuaris els sistemes de fitxers local i remots.
 - Monitoritzar i controlar els recursos de disc.
 - Protecció davant d'errors hardware, seguint polítiques adequades de back up.
 - Limitar l'accés per garantir la confidencialitat de les dades.
 - Comprovar periòdicament la integritat del sistema.
 - Afegir nous dispositius d'emmagatzemament quan sigui necessari.

Sistemes de Fitxers: Introducció

- Pràcticament TOT en els entorns UNIX està mapejat al sistema de fitxers. Exemples:
 - Els dispositius
 - Exemple:
 - » `cat > /dev/tty`
 - » `cat /dev/psaux`
 - El proc filesystem, que permet consultar on-line l'estat del sistema operatiu, dispositius associats, i paràmetres de configuració.
 - Exemple:
 - » `cat /proc/pid/mem`
 - `/var/log`: Logs del sistema.

Sistemes de Fitxers

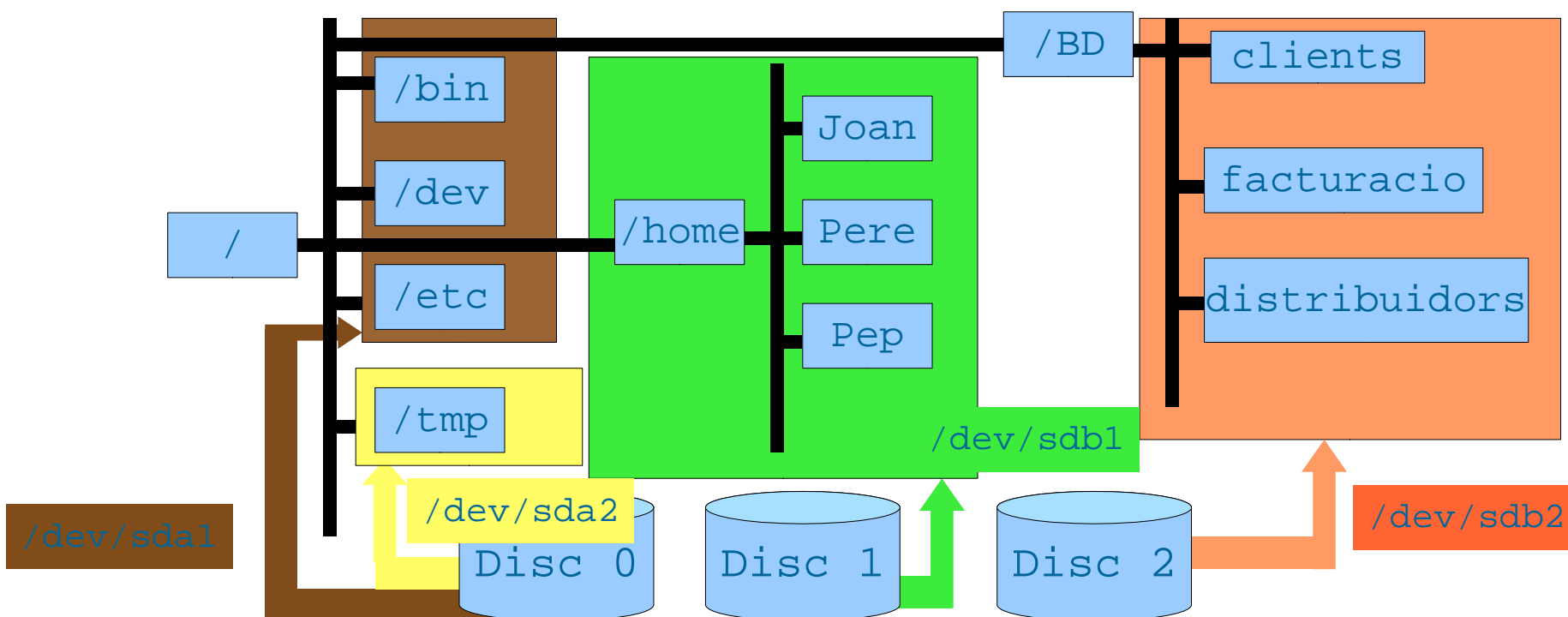
- Per a poder fer servir un disc o una partició del disc, es necessari crear-hi un sistema de fitxers.
 - Mantenen les estructures de dades necessàries per a permetre l'accés als fitxer lògics desde els sectors físics.
- **Mida de bloc:** És l'espai unitari d'emmagatzemament de disc.
 - No es pot direccionar res inferior a aquesta mida, i tot el que ocupi menys implica perdre l'espai restant.
 - Mida petita: Ideal per a sistemes amb molts fitxers petits.
 - Mida gran: Ideal per a sistemes amb fitxers grans.
- Exemples de sistemes de fitxers: fat32, nfs, swap, iso9660 (CD), ext2, **ext3**

Sistemes de Fitxers

- Sistemes de fitxers transaccionals: **Journalized filesystems.**
 - Els sistemes de fitxers tradicionals (basats en BSD) són lents alhora de comprovar la integritat del sistema.
 - Solució: Guardar metadades dels canvis que es fan a disc en un log, seguint un model transaccional.
 - Si el sistema es penja a mig efectuar una operació, les transaccions no acabades es repeteixen, garantint la integritat del sistema.
 - Permet combinar eficientment diferents operacions d'escriptura.

Sistemes de Fitxers

- Operacions sobre el sistema de fitxers: Muntar i desmuntar.
 - Muntar un sistema de fitxers és fer accessible a la resta del sistema un arbre de directoris comú.



Sistemes de Fitxers

- Operacions sobre el sistema de fitxers: Muntar i desmuntar.

- Muntar:

```
mount <dispositiu> <tipus> <opcions> <directori -  
punt de muntatge>
```

Opcions: r (read-only), w (write), ...

Exemples:

```
mount /dev/hdb1 -t ext3 /home
```

```
mount /dev/cdrom /cd -t iso9660 -o ro,user,noauto,unhide
```

- Desmuntar:

```
umount <directori> o <dispositiu>
```

Sistemes de Fitxers

- Noms de dispositiu:
 - Convenció: S'usa una arrel pel tipus del dispositiu, una lletra per indicar el dispositiu concret, i un número que indica la partició. Exemple:
 - /dev/**hda**1: primera partició del primer disc ide.
 - /dev/**hdb**3: tercera partició del segon disc ide.
 - /dev/**sdc**2: segona partició del tercer disc SCSI.
- No es permet desmuntar un sistema de fitxers si està en ús.
- Utilitats:
 - **fuser**: mostra tots els usuaris i processos que estan accedint a un determinat recurs. Exemple:
 - `fuser -u /dev/sda1`
 - `fuser -k readme.txt` (mataria els processos que usen el fitxer)
 - **lsuf**: mostra una llista de tots els fitxers oberts.

Sistemes de Fitxers

- Configuració dels sistemes de fitxers en temps de boot:
/etc/fstab.
 - Fitxer on hi ha una llista dels sistemes de fitxers reconeguts que es muntaran en arrencar l'ordinador.
 - Format:
<sist. fit.> <punt muntatge> <tipus fs> <opcions> <freq-dump> <fsck-pass>
<sis. Fit.>: fitxer especial del sistema de particions (/dev/sda1...)
<punt muntatge>: Directori d'accés des del sistema de fitxers.
<tipus fs>: nfs, ext3, vfat, iso9660,...
<opcions>: rw, ro, noexec, noauto (no muntar en boot, ex: cdrom),...

Sistemes de Fitxers

<freq-dump>: freqüència en la que el sistema de fitxers serà sotmès a un procés de back up. 0 indica no back up.

<fsck-pass>: Ordre en el que es comprova la integritat del sistema de fitxers.

Exemple:

| | | | | | |
|------------|---------------|------|---------------------|---|---|
| /dev/hda2 | / | ext2 | defaults | 1 | 1 |
| /dev/hdb1 | /home | ext2 | defaults | 1 | 2 |
| /dev/cdrom | /media/cdrom | auto | ro,noauto,user,exec | 0 | 0 |
| /dev/fd0 | /media/floppy | auto | rw,noauto,user,sync | 0 | 0 |
| proc | /proc | proc | defaults | 0 | 0 |
| /dev/hda1 | swap | swap | pri=42 | 0 | 0 |

- Muntar automàticament tots els sistemes de fitxers definits a /etc/fstab:
mount -a

Sistemes de Fitxers

- Validació de la integritat del sistema de fitxers:
 - Problemes: El sistema pot caure, penjar-se o produir-se una fallida HW. El sistema de fitxers pot quedar en una situació inconsistent.
- Utilitat **fsck**: Comprova la coherència del sistema de fitxers comparant la llista de blocs lliures amb els blocs guardats als i-nodes.
 - Permet la correcció de problemes (blocs marcats com a lliures que estan en ús, ...).
 - No pot recuperar dades corruptes.

```
fsck [opcions] dispositiu
```
- Després de canvis en el root filesystem, cal fer un reboot.

Sistemes de Fitxers

Fitxers i arxius en UNIX: utilitats

- Comandes elementals per a l'accés i manipulació de fitxers en UNIX:

- Crear, eliminar i accedir a directoris:

```
>mkdir directori    >rmdir directori    >cd directori
```

- Conèixer el directori actual i mostrar-ne el contingut:

```
>pwd                >ls <opcions> <directori>, ls -la
```

- Moure/renombrar fitxers directoris:

```
>mv directori_antig directori_nou
```

- Copiar directoris:

```
>cp -rR origen destí (recursívament els subdirectoris)
```

Sistemes de Fitxers

Fitxers i arxius en UNIX: utilitats

- Comandes elementals per a l'accés i manipulació de fitxers en UNIX:
 - Creació de fitxers:
>touch fitxer
 - Visualitzar el contingut de fitxers:
>cat fitxer >more fitxer >less fitxer
>tail -20 fitxer >head -20 fitxer
 - Copiar i esborrar fitxers:
>cp origen desti
>rm fitxer

Sistemes de Fitxers

Fitxers i arxius en UNIX: utilitats

- Enllaços (links):

- Permeten que un fitxer faci referència a un altre fitxer:

- **Hard links:** Associen a dos fitxers el mateix inode de disc (compartit)

```
>ln fitxer hlinkfitxer
```

- **Soft links (enllaços simbòlics):** Fitxers que fan referència a altres fitxers, apuntant a la seva ruta.

```
>ln -s fitxer slinkfitxer
```

Diferència: En esborrar el fitxer, no podrem accedir-hi mitjançant slinkfitxer. Sí que hi podríem accedir mitjançant hlinkfitxer, ja que només s'esborra realment quan el comptador d'enllaç dels inodes arriba a 0.

Sistemes de Fitxers

- **Volums Lògics LVM (Logical Volume Managers):**
 - Conceptualment diferent al sistema de particions.
 - Permet que un sistema de fitxers ocupi més que un disc.
 - Permet augmentar la mida de sistemes de fitxers sense haver-lo de crear de nou.
 - Permet mirroring per protecció de dades (copia-redundància).
 - Permet combinar diferents discs en un sol disc lògic (disc stripping).
- **Avantages:**
 - Podem unir diferents discs físics en un volum lògic i redimensionar cada partició segons les nostres necessitats.

Sistemes de Fitxers

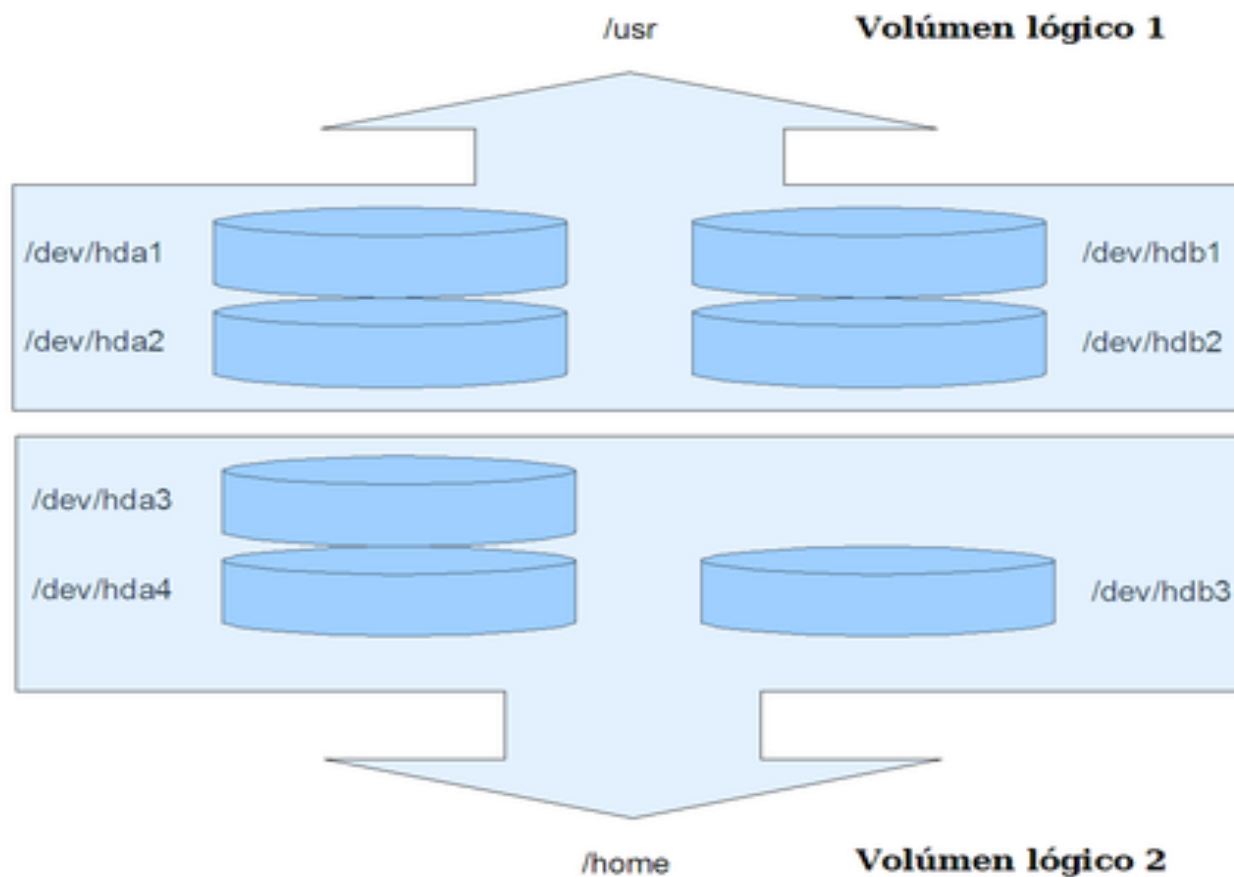
- **Volums Lògics LVM.**
- Elements:
 - Volums físics o discs: El hardware físic que guarda la informació.
 - Volums lògics: Dispositiu estàndard visible. Similar a una partició UNIX, on es podran crear sistemes de fitxers.
 - Grup de volums: Agrupació de discs i volums lògics. Poden estar subdividits en grups de volum físics.
 - Particions físiques (*extents*): Unitats d'emmagatzemament que pot reservar espai.
 - Particions lògiques (*logical extents*): Unitats d'emmagatzemament lògic.

Sistemes de Fitxers

- **Volums Lògics LVM.**
- Normalment una partició lògica té associada una partició física.
 - En cas de voler guardar dades amb redundància, una partició lògica es pot correspondre a més d'una partició física, guardant dos o més còpies de cada bloc (mirrors)
- Si es vol incrementar la mida d'un sistema de fitxers, només cal incrementar el nombre de particions físiques associades.
- Reduir la mida d'un sistema de fitxers implica treure particions físiques associades.

Sistemes de Fitxers

- Volums Lògics LVM.
- Esquema:



Sistemes de Fitxers

- **Volums Lògics LVM.**
- **Disc striping:** Augmentar el rendiment del sistema de fitxers, ajuntant varis discs en un volum lògic:
 - Augmenta la capacitat del sistema
 - Augmenta la velocitat, ja que es poden atendre diverses peticions en paral·lel.
 - Instal·lar els discs en controladors diferents
- **RAID (*disc mirroring*):** Redundant Array of Inexpensive Disks). Consisteix en crear còpies redundants de les dades del disc per incrementar la seguretat en cas de fallida d'un dels dispositius.

Sistemes de Fitxers

- **Volums Lògics LVM.**
 - **RAID:** En funció de quants i com s'organitzen els discs redundants es defineixen diferents nivells de RAID:
 - Raid 0: Sense tolerància a errors, unió directa de dos discs (striping).
 - Raid 1: Cada disc en té un altre on hi ha la informació duplicada.
 - Raid 2: Cada N discs tenen associat un altre on hi ha redundància en forma de codi corrector de Hamming(*)
 - Raid 3: Cada disc té associat un bit de paritat.
 - Raid 4: Discs individuals amb un disc de paritat a nivell de bloc.
- (*) Sistema usa en telecomunicacions per detectar i corregir errors en trames de bits

Sistemes de Fitxers

- **Volums Lògics LVM.**

- Raid 5: Disc amb paritat a nivell de bloc, on la informació de paritat es troba distribuïda de forma intercalada entre els discs.
- Raid 6: Discs amb un doble nivell de paritat.
- Raid 10: Combinació de RAID 1 + 0. Consisteix en un conjunt de discs RAID 0, que internament consten de un RAID 1 (concatenació de mirrors)
 - Raid 0+1: Mirror de discs en raid 0.

Més informació: <http://es.wikipedia.org/wiki/RAID>

Sistemes de Fitxers

- **Volums Lògics LVM.**
- Exemple: Linux Logical Volume Manager
 - Cal recompilar i inserir els mòduls corresponents al kernel
 - Afegir als scripts d'inici i apagat l'activació i desactivació de LVM:
vgchange -a y i vcchange -a n
 - Comandes:
 - Volums físics: pvcreate, pvdisplay, pvchange, pvmove ,pvscan.
 - Volums lògics: lvcreate, lvdisplay, lvchange, lvremove, lvreduce, lvextend, lvscan, lvrename.
 - /etc/lvmtab: paràmetres de configuració actuals.

Més informació:

LVM: <http://www.tldp.org/HOWTO/LVM-HOWTO/>

RAID: <http://linas.org/linux/raid.html>

<http://bulma.net/body.phtml?nIdNoticia=1863>

Sistemes de Fitxers: floppy

- Accés als dispositius de disc tipus floppy:
 - S'hi accedeix muntant els dispositiu: /dev/fd0.
 - Donar format a un disc MS-DOS(*):

```
fdformat /dev/fd0
mkfs -t msdos /dev/fd0
mount /dev/fd0 /mnt
```
 - **mttools**: Conjunt d'utilitats que permeten accedir a sistemes DOS de manera similar a com es feia al DOS original:

```
mformat a:, mcopy *.* a:, mdir, mtype, mcd, mlabel, mdel,
mren, mmd, mrd, mattrib.
```

Es corresponen a les comandes copy, dir, type, cd, label, del ren, md, rd i attrib del DOS original.
- Accés al CDROM: similar al floppy. Dispositiu /dev/cdrom.

Sistemes de Fitxers

- Compartició de fitxers:
 - Existeixen dues maneres de compartit fitxers:
 - Entre dos entorns UNIX: NFS (Network File System)
 - Entre UNIX i entorns Windows (Samba)
- NFS
 - Permet que sistemes de fitxers residents físicament en una màquina siguin accessibles per altres màquines com si es tractés d'un disc local més (NAS, **SAN**).
- Samba
 - Suporta el protocol Server Message Block (SMB), natiu de Windows, permeten que els recursos UNIX es vegin com recursos compartits estàndard de Windows.

Sistemes de Fitxers

- **NFS** : Aspectes de configuració

- Cal definir a `/etc/fstab` una entrada per cada sistema remot que es vol incorporar.

```
#Dispositiu          muntatge  tipus  opcions  dump  fsck
maquina1:/carpeta /carpetanfs nfs      bg      0      0
```

- Per oferir carpetes compartides cal definir al `/etc/exports`. Exemple:

```
#Carpeta      opcions
/carpeta      maquina2(rw) maquina3(ro)
```

el directori carpeta podrà ser llegit i escrit per la màquina 2, mentre que la màquina 3 només podrà llegir-hi.

- Cal tenir instal·lats els daemons dels serveis de xarxa NFS. Per més detalls veure la secció NFS del Tema "Serveis de xarxa".

Sistemes de Fitxers

- **Samba** : Aspectes de configuració

- Cal definir a `/etc/samba/smb.conf` una entrada amb informació de cada recurs que es vol exportar.

`smbstatus`: permet coneixer els usuaris actuals dels recursos compartits.

- Client de samba: Els paquets de samba incorporen utilitats per a accedir a sistemes Windows, normalment proporcionant un tipus de sistema de fitxers per al seu muntatge. Exemple:

```
mount -t smbfs -o username=usuari, password=***  
//maquina_windows/recurs_windwos /carpeta_local
```

- Cal tenir instal·lats els daemons dels serveis de xarxa SMB. Per més detalls veure la secció Samba del Tema "Serveis de xarxa".

Gestió del espai: quotes

- Serveixen per a limitar la quantitat màxima d'espai de que podrà disposar cada usuari.
 - Per cada sistema de fitxers i usuari el sistema operatiu manté l'espai consumit i restant.
 - Dos tipus de quotes:
 - Limit dur (hard limit).
 - Limit tou (soft limit).

Gestió del espai: quotes

- No es permet als usuaris sobrepassar MAI el limit dur. Qualsevol operació d'escriptura fallarà al reservar espai.
- Es permet que superin el limit tou de forma temporal.
 - S'envien missatges d'avís a l'usuari cada cop que es connecta al sistema.
 - Fins que redueix el consum d'espai per sota del limit tou.
 - Fins que es advertit un cert número de vegades, negant futures reserves d'espai fins que no elimini fitxers per sota del limit tou.

Gestió del espai: quotes

- L'administrador ha de decidir quins sistemes de fitxers requereixen de quotes:
 - Normalment solen ser els ocupats pel home dels usuaris.
 - **apt-get install quota**
- Preparació del sistema de fitxers: Cal editar i indicar a les opcions que el sistema de fitxers usa quotes:

```
/dev/sda1 /home ext3      usrquota,grpquota 1 1
```
- Cal desmuntar i muntar de nou el sistema de fitxers després de fer els canvis en **/etc/fstab**.

Gestió del espai: quotes

- Assignació de l'espai limit:

> edquota usuari

S'encarrega de editar les propietats de la quota d'un determinat usuari:

```
/sistema blocs in use: 14578, limits (soft=30000), hard=35000)  
        inodes in use: 843, limits (soft=0), hard=0)
```

Per a desactivar les quotes només cal especificar el valor 0.

- Establir el temps màxim d'expiració de la soft quota:

>edquota -t

```
/sistema: block grace period: 3days, file grace period: 0 days
```

- Fitxers de quota:

– Per cada partició amb quota incloure els fitxers a l'arrel:

- aquota.user
- aquota.group

Gestió del espai: quotes

- Habilitar/deshabilitar les comprovacions de quota:
>quotaon sistema_fitxers
>quotaoff sistema_fitxers
- Comprovar la consistència (després de modificacions):
>quotacheck -a (tots els sistemes de fitxers)
- Informes de l'ús actual de les quotes per cada usuari:

```
>repquota -v /system
```

| | Bloc limits | | | | file limits | | | |
|--------|-------------|-------|-------|-------|-------------|------|------|-------|
| User | used | sof | hard | grace | used | soft | hard | grace |
| Usuari | 14578 | 30000 | 35000 | 3 | 843 | 0 | 0 | |

Gestió del espai: quotes

- Automatitzar el control de quotes:
 - `/etc/crontab` per setmana:
 - `0 3 * * 0 /usr/sbin/quotacheck -avug`
 - Resultats a `aquota.user`, `aquota.group`.

Impressores i cues d'impressió

- En els entorns UNIX, existeix una gran varietat de sistemes d'impressió:
 - Sistema BSD: Linux, FreeBSD i True64.
 - System V: Solaris, i HP-UX.
 - AIX utilitza el seu propi sistema d'impressió.
- Components principals:
 - Impressores: laser, tinta, USB, paral·lel, xarxa,...
 - Comandes d'impressió.
 - Cues d'impressió: guarden els treballs d'impressió.

Impressores i cues d'impressió

(Components principals)

- Spooling directories: Guarden els treballs pendents
- Processos de servidor: Accepten, guarden i envien treballs a imprimir al dispositiu.
- Filtres: Transformen els fitxers al format adequat pel dispositiu.
- Comandes d'administració: Engegar/parar el sistema d'impressió, gestionar les cues, els treballs i els paràmetres dels dispositius.
- Impressió remota: servidor d'impressió que rep peticions de molts clients.

Impressores i cues d'impressió

- Comandes d'usuari

- Basades en el daemon lpd: capaç de mantenir múltiples cues i impressores, locals i remotes.
- lpr: permet enviar un treball a imprimir. Lpd assigna un Identificador al treball, que s'usa en totes les comandes, assegura disponibilitat i filtra el treball per impressió.
- lpq: llista els treballs que hi ha a la impressora.

```
>lpq -P impressora
```

- lprm: elimina un treball de la cua d'impressió.

```
>lprm -P impressora 13 (elimina el treball 13 de la impressora)
```

- arxiu **printcap** --> fa el control de lpr i lpd.

Impressores i cues d'impressió

- Cues d'impressió

- El llistat de dispositius acceptats pel sistema: `/etc/printcap`

- Exemple:

```
# Local ASCII printer
```

```
lp1|printer
```

Nom i alias

```
:server
```

```
:cm=Dumb printer
```

```
:lp=/dev/lp1
```

Dispositiu a enviar el treball

```
:sd=/var/spool/lpd/lp1
```

Spool directory

```
:lf=log:af=acct
```

fitxers de log i accounting

```
:filter=/usr/local/libexec/filters/ifhp
```

programa filtre

```
:mx=0
```

mida màxima del treball (0 = NO)

- Més informació: <http://www.lprng.com/LPRng-HOWTO-Multipart/cm.htm>

Impressores i cues d'impressió

- Spooling directories
 - Guarden els treballs per imprimir fins que lpd els pot processar.
 - Cada impressora té un spooling directory a:
`/var/spool/lpd/impressora`
 - Creació d'un nou spooling directory:
 - `>cd /var/spool/lpd`
 - `>mkdir novaImpressora`
 - `>chown daemon.daemon novaImpressora`
 - `>chmod 755 novaImpressora`
- Spooling daemon: `/usr/sbin/lpd`
 - Només es permet una instància de lpd en funcionament:
`/etc/init.d/lpd restart`

Impressores i cues d'impressió

- Filtres:
 - Utilitzen comandes i utilitats que els permeten filtrar i preparar els documents per a la impressora.
 - Guarden informació d'accounting (número de pàgines impreses,...)
 - Utilitats:
 - Ghostscript: permeten imprimir un fitxer PostScript en una impressora no PostScript (*)
 - a2ps: conversor de ASCII a PostScript.
 - Ghostview: utilitat per a visualitzar fitxers PostScript.
 - Mpage: Utilitat per imprimir més d'una pagina per full.

(*)PostScript: Llenguatge de descripció de pàgina usat en la majoria diimpressores.

Impressores i cues d'impressió

- Comandes de gestió i administració de les cues d'impressió
 - Line printer control utility: lpc.
 - Comandes internes:
 - Status impressora: Mostre l'estat de la cua de la impressora.
 - Stop impressora: Para la impressió quan acabi el treball actual.
 - Abort impressora: Elimina els treballs i deshabilita la impressió.
 - Clear impressora: Elimina tots els treballs de la cua.
 - Start impressora: Reengega la impressió després d'un stop o abort.
 - Desable impressora: Evita que es puguin enviar treballs.
 - Enable impressora: Permet que es tornin a enviar treballs a la cua.
 - Down impressora: para la impressió i desactiva la cua.
 - Up impressora: activa la cua i comença la impressió.

Impressores i cues d'impressió

- Impressió windows filtra/imprimeix de forma simultànea.
- Sistemes UNIX esperen que tota la informació es filtri de forma correcte:
 - Evita problemes d'espai i de pèrdua de tasques a mig processar.

Impressores i cues d'impressió

- Impressió remota

- Les impressores remotes en xarxa suporten directament el protocol d'impressió LPD (podem canviar la informació local a remota en **printcap**).

```
remotalp|printer
```

```
:lp=: \
```

Indica que es remota

```
:rm=maquina_remota:rp=impressora: \ Host on s'enviarà
```

```
:sd=/var/spool/lpd/lp1
```

```
:lf=log:af=acct
```

- Defineix una impressora remota anomenada remotalp.
- Per a que una màquina remota accepti treballs d'impressió, cal definir a **/etc/hosts.lpd** la llista dels hosts remots que tenen permís per imprimir.

Impressores i cues d'impressió

- Afegir una impressora
 1. Connectar la impressora
 2. En el cas d'impressores sèrie, comprovar que la línia es troba disponible
 3. Verificar que el procés lpd esta iniciat, i existeixen els scripts de boot.
 4. Afegir una entrada a /etc/printcap per la impressora.
 5. Crear un spooling directory.
 6. Crear un fitxer de accounting:

```
touch /var/adm/lpd_acc/impressora
```

 1. Inicialitzar la impressora: `lpc up impressora`
 2. Imprimir una pàgina de test.

Impressores i cues d'impressió

- Millores del sistema d'impressió lpd:
 - LPRng
 - Paquet que es una extensió de LPD amb noves característiques afegides.
 - Compatible amb LPD (lpr, lpq, lprm,...).
 - Permet l'ús de prioritats d'impressió.
 - Classes d'impressió: útil per treballs que requereixin un tipus de paper especial.
 - Connexió amb LDAP directory.
 - Control d'accés.

Impressores i cues d'impressió

- Millores del sistema d'impressió lpd: CUPS
 - CUPS: Common Unix Printing System.
 - Específicament dissenyat per imprimir en xarxa.
 - Implementat seguin el Internet Printing Protocol (IPP), implementat sobre http. CUPS requereix un servidor web al servidor d'impressió. `http://localhost:631`
 - Suporta compatibilitat amb `lpr`, `lpq`, `lprm`, etc.
 - Afegeix comandes com `lpadmin`, per afegir i configurar noves impressores.
- ```
>lpadmin -plj4 -D"Impressora informàtica" -L"Sala B-342"
-vsocket://192.168.4.5 -mlaserjet.ppd
```
- `arxiu.ppd`, arxiu driver amb les dades per defecte d'impressió, etc.
  - Veure estat d'una cua: **`lpstat -t`**

# Impressores i cues d'impressió

- Millores del sistema d'impressió lpd: CUPS
  - Fitxers de configuració a `/etc/cups`. Es mantenen amb `lpadmin`.
  - Fitxer global de configuració: `cupsd.conf`: control d'accès i propietats.
  - Té control d'accés i autenticació.
  - Tot el control CUPS d'impressió per http port 631
  - Més informació: [www.cups.org](http://www.cups.org)

# Entorn Gràfic: X

- El entorn gràfic X-Windows proporciona la interfície gràfica al món LINUX.
- Es basa en un model **Model client-servidor**.
  - La màquina on s'executa l'entorn X es independent de la màquina on es fa visualització.
  - Podem exportar la sessió gràfica X, i remotament, treballar com si estiguéssim davant del monitor de la nostra màquina.

# Entorn Gràfic: X

- Entorns de treball: Permeten una GUI més coherent i amigable per sobre del sistema X base, proporcionant mecanismes addicionals de cohesió entre aplicacions
- Hi ha dos grans entorns competidors:
  - KDE (K Desktop Environment).
  - GNOME.

# Entorn Gràfic: X

- Configuració de l'entorn X-Windows:
  - Antic: Eines específiques: **Xconfigurator**. Permeten l'autodetecció del model de monitor, hardware de vídeo, i modes de resolució disponibles. (**Xf86config**: script amb tots els paràmetres de configuració).
  - Actualment: XORG -> /etc/X11/xorg.conf
  - Engegar les x-windows: comanda `startx`

# Entorn Gràfic: X

- Fonts i Entorns X
  - Dos grans famílies de fonts:
    - Fonts bitmap: Es guarden en una imatge.
    - Fonts vectorials: Es guarden com una expressió matemàtica de les corbes que les generen. Son infinitament escalables.
  - Les font vectorials es divideixen en dues subfamílies:
    - Adobe Type 1: Usen una codificació basada en curves de Bezier.
    - TrueType: Usen una codificació basada en b-splines.
  - Normalment les fonts es guarden a `/usr/X11R6/lib/X11/fonts`
  - Per a la seva impressió, es necessari un preprocés per part d'una utilitat específica: Ghostscript ([www.ghostscript.org](http://www.ghostscript.org))



# Bash 1

Administració de Sistemes Operatius  
Dept. Matemàtica Aplicada i Anàlisi,  
Universitat de Barcelona

# Index

- Introducció
- Variables
- Control de la shell
- Expressions regulars
- Pipes/redireccionament
- Comandes
- Operacions

# Introducció

- **bourne-again shell**
- Bash és un shell de UNIX (intèrpret de comandes) com el tradicional Bourne-Shell
- Es va definir per al projecte GNU

# Introducció

- Arxius de bash
- /bin/bash --> executable de bash
- /etc/profile --> iniciació usada per la shell
- ~/.bash\_profile --> inicialització personal
- ~/.profile --> emprat per la shell
- ~/.bash\_login --> executa quan entra a la shell
- ~/.bash\_logout --> executa quan surt de la shell
- ~/.bashrc --> arxiu personal d'inicialització de la shell
- ~/.inputrc --> arxiu d'inicialització individual

# Variables

- `variable=valor`
- `$variable`
  - Mostra el contingut de la variable
- `$?`
  - Retorna la salida de l'última comanda executada
  - Valors de retorn:
    - Depèn de la comanda executada. Estándar:
      - 0: sense errors
      - 127: la comanda no existeix
      - 1: error variable
  - `man comanda`: per veure el tipus de retorn de una comanda

# Variables

- $\$variable$  ó  $\${variable}$  --> contingut de la variable
  - $\${variable}2$  --> concatenació del contingut amb el número 2.
- Comanda paràmetre1 ... paràmetreN:
  - -->  $\$# = N+1$ 
    - $\$0$  : comanda
    - $\$1$  : paràmetre1
    - ...
    - $\$N$  : paràmetreN
- $\$USER$  --> nick del user
- $\$$$  --> PID del procès en execució
- $\$PPID$  --> PID del procès pare

# Variables

- `$*` --> retorna paràmetres separats per IFS
- `$@` --> retorna paràmetres com a llista de paraules  
"`$1`" "`$2`"...
- `$_` --> resultat de l'anterior comanda executada
- `#!` --> identificació del procés de l'última comanda que va començar amb `&`
- `date`
  - --> `jue mar 1 12:42:17 CET 2007`
- `set variable=valor` --> variables locals
- `setenv variable valor` --> variables d'entorn

# Variables

- Paràmetres: \$1 \$2 \$3 \$4 \$5 \$6 \$7 \$8 \$9
- Per accedir al paràmetre 10 no existeix \$10:
  - » shift:
    - \$1<--\$2
    - \$2<-- \$3
    - \$3<-- \$4
    - \$4<-- \$5
    - \$5<-- \$6
    - \$6<-- \$7
    - \$7<-- \$8
    - \$8<-- \$9
    - \$9<-- 10
  - » shift n --> desplaçament de n posicions



# Variables

- Variables locals:
  - history --> comandes anteriors
  - home --> directori quan fem cd
  - pwd --> directori actual
  - path --> directoris on es busquen recursos (separats per espais)
  - status --> estat de l'última comanda (0 correcte)

# Variables

- Variables globals:
  - PATH --> directoris on es busquen recursos (separats per ':')
  - TERM --> indica tipus de terminal
  - PRINTER ó LPDEST --> mostren cua d'impressió per defecte
  - DISPLAY --> apareixen les sortides de X-Windows
  - HOSTNAME --> nom de la màquina
  - MAIL --> arxiu de correu
  - SHELL --> intèrpret de comandes per defecte
  - USER --> nom de l'usuari

# Variables

- history:

- » !numero --> executa comanda amb identificador 'numero'
- » !! --> última comanda
- » !-n --> comanda 'n' desde el final
- » !cadena --> última comanda que comenci per 'cadena'
- » ^viejo^nuevo --> última comanda substituint 'viejo' per 'nuevo'

# Variables

- Variables del Bourne Shell (es poden llistar les definides per set)
  - HOME
  - pwd
  - PATH (separació ':')
  - PS1 --> cadena que imprimeix quan espera comandes
  - PS2 --> prompt secundari
  - **IFS** --> Internal Field Separator (caràcter per separar arguments a la línia de comandes, espai per defecte).
  - TERM

# Variables

- Assignació
  - ' --> l'expressió s'agafa de forma exacta sense interpretació del contingut
  - " --> Com l'anterior però substitueix el tipus \$variable
  - ` --> El contingut s'interpreta com una comanda

# Variables

- Variable estructura

- » `tab[1]=hola`
- » `tab[2]=adios`
- » `tab[index]=atribut`
  
- » # Accedir al contingut:
- » `${tab[index]}`
  
- » `echo ${tab[*]}`
- `hola adios`

# Control de la shell

- Execució script:
  - `bash script.sh`
  - `sh script.sh`
  - `chmod +x script.sh ; ./script.sh`
- Debug:
  - Emprar `echo` en tots els llocs possibles
  - `bash -n -->` per errors de sintaxi
  - `bash -v -->` verbose dump per a cada línia que llegueix el shell (`+v` per finalitzar)
  - `bash -x -->` per veure que executa cada comanda (`+x` per finalitzar)

# Control de la shell

- **Sub-shell**

- » `var1=888`
- » `(var1=111;echo $var1)`
  - 111
  - `echo $var1`
  - 888

- **Funcions:**

- » `#!/bin/bash`
- » `function (){`
- » `codi`
- » `}`



# Control de la shell

- Funcions: valor de retorn
  - » return X
  - » exit (== exit \$? ) --> allibera recursos

# Control de la shell

- **Àmbit:** les variables dins d'una funció pertanyen a tot l'àmbit de la shell activa:

```
#####
```

```
function1() --> FUNCIO
```

```
{
```

```
echo '<$1>'
```

```
echo '('$2')'
```

```
var=1234
```

```
}
```

```
function1 111 222 --> MAIN
```

```
echo $var
```

```
#####
```

```
--> <111>
```

```
 (222)
```

```
 1234
```

# Control de la shell

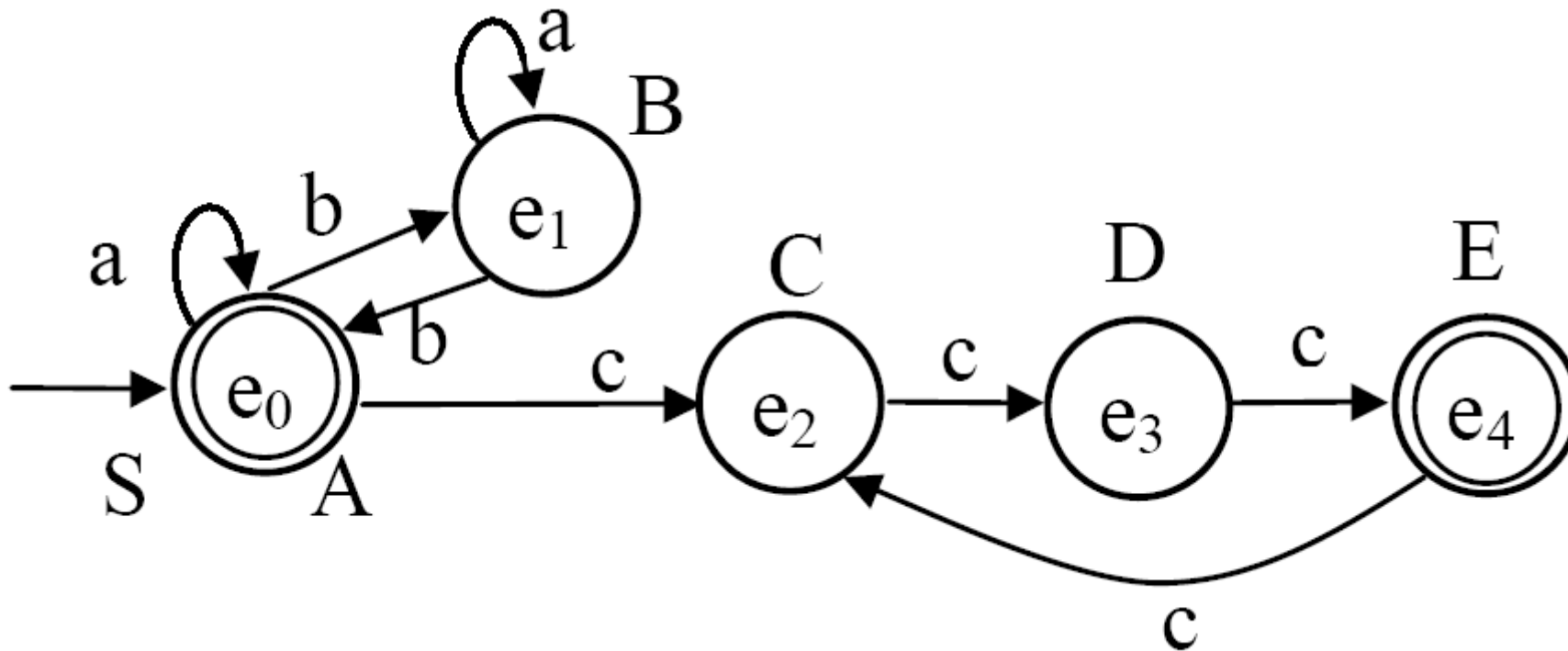
- **Inputs:**
  - **read** variable1 ... variableN --> espera entrada per teclat i assigna el contingut a les variables
- **eval** expressió --> evalua l'expressió
- **echo -n** "informació" --> mostra informació sense un intro

# Expressions regulars

- O també **patró**
- **Regles** que defineixen un llenguatge regular
- Relacions entre alfabet d'entrada --> llenguatge

# Expressions regulars

- Es poden representar amb un autòmat finit



# Expressions regulars

- \*  
a\*b --> 0 ó més 'a' i després 'b'
- +  
a+b --> 1 ó més 'a' i després 'b'
- .  
a.b --> 'a' i una unitat de qualsevol cosa i 'b'
- ? --> qualsevol caràcter 0 o més cops.
- {,}  
. {1,2} --> 1 ó 2 unitats ({1,2}) de qualsevol cosa (.) (p.e. P{3} --> PPP)

# Expressions regulars

- $\wedge$  (inici)
  - $\wedge a$  --> una 'a' a l'inici de fila
- $\$$  (fi)
  - $b\$$  --> una 'b' al final de línia
- $\<$  --> patró al principi de paraula
- $\>$  --> patró al final de paraula
- $\backslash b$  --> patró al principi ó final de paraula
- $\backslash B$  --> patró ni al principi ni al final de paraula

# Expressions regulars

- [-]
  - [a-c] --> 1 unitat que pot ser 'a', 'b' ó 'c'
  - [a-zA-C0-9] --> OR!! 1 unitat que pot ser:
    - ó un caràcter entre 'a' i 'c', ó un caràcter entre A i C, ó un número entre 0 i 9 (sense espais ja que l'espai també seria una possible instància) (utilitat: [^simbol] --> que no aparegui el símbol)
- | --> OR
- [a|b] --> ha de coincidir amb un dels caràcters de qualsevol costat de la barra
- (XY) --> ha de coincidir, considera tota la cadena com un àtom
- (p.e. a(X|Y) --> aX, aY)



# Expressions regulars

- Fer una expressió regular per a qualsevol expressió que **contingui** 'paraula' i just després hi **hagi** com a mínim un número.
- Fer una expressió regular per a qualsevol expressió que **comenci** per 'paraula' i just després hi **hagi** una lletra.
- Fer una expressió regular per a qualsevol expressió que **comenci** per 'paraula', que **contingui** com a mínim **un 3**, i que **acabi amb 'F'**.
- Fer l'expressió regular de l'autòmat de la dispositiva 21

# Pipe (canonada)

- Tota comanda té 3 canonades:
  - Entrada estàndar: teclat
  - Sortida estàndar:
    - Sortida estàndar pantalla
    - Sortida error pantalla
- Es preferible petites eines per resoldre problemes, però necessitem ensamblar aquestes eines en estructures més complexes
  - --> redireccionament entrada/salida

# Pipe (canonada)

- `a > b` sortida estàndar de 'a' es posa a 'b' (si 'b' existeix s'esborra abans)
- `a 2 > b` sortida d'error de 'a' es posa a 'b' (si 'b' existeix s'esborra abans)
- `a &> b` sortida estàndar i d'error de 'a' es posa a 'b' (si 'b' existeix s'esborra abans)
- `a >> b` sortida de 'a' es posa al final de 'b'
- `comanda < fitxer` La comanda agafa la seva entrada d'un fitxer
- `a ; b` s'executa 'a' i després 'b'
- `a | b` s'executa 'a' i la seva sortida és l'entrada de 'b'

# Comandes

- Algunes comandes importants
- tr
- cut
- grep
- wc
- sed
- head
- tail
- cat
- **awk -> llenguatge propi**

# Comandes

- tr

```
#echo hola | tr 'a' 'z'
```

```
holz
```

```
#tr -s 'X'
```

substitueix tota una seqüència de X per només una X

```
#tr -d 'X'
```

esborra totes les ocurrències de X

– 'A-Z' --> de la 'A' a la 'Z'

– 'b-za' --> de la 'b' hasta la 'a'

– 'd-za-c' --> de la 'd' a la 'c'

# Comandes

- Emprar **tr** per a generar la codificació i decodificació de la criptografia de Cesar (**suma de tres caràcters per posició**):  
(p.e hola --> codificació --> ipmb --> decodificació --> hola)

# Comandes

- cut

`#echo 'a;b;c' | cut -f1 -d \;` (**\ ja que ; és un caràcter especial**)

a --> primer camp amb separador ';', si no es posa separador per defecte busca tabulador

-f1-3 --> els camps de 1 al 3

-b2-4 --> agafa per bytes

Bona pràctica: `tr + cut` (per ajustar separador abans del cut)

# Comandes

- grep

#busca ocurrències

opció -E (extended, permet fer ús de {},+,etc.)

cat arxiu | grep -v paraula

--> Totes les línies que **no** continguin 'paraula'



# Comandes

- WC

#

-l línies

-c caràcters

-w paraules

# Comandes

- sed (-r per tot el conjunt d'expressions regulars)

```
#sed 's/hola/adios/g'
```

s --> substituir

hola --> paraula origen

adios --> paraula destí

/g --> per a totes les coincidències de línia  
(sinó només canvia la primera de cada línia)

```
#sed -e '/X/!d'
```

Esborra tot excepte les línies que continguin 'X'

```
#sed -e operació1 -e operació2 -e operació3 ...
```

S'executen de manera consecutiva les  
operacions si les anteriors s'han executat  
correctament

# Comandes

- sed

sed "s/.\*\ (ho\)/\1la/g" fitxer

--> canvia totes (/g) les ocurrències de 'ho' per hola, ja que fem substitució dels primers \(\) al fitxer

p.e: fitxer = hola bon dia:

--> holala bon dia

# Comandes

- head
  - Mostra línies inicials (per defecte 10)
- tail
  - Mostra últimes línies (per defecte 10)
  - #tail -2 nom\_fitxer
    - Mostra les dues últimes línies
  - #tail +2 nom\_fitxer
    - Mostra desde la segona fins l'última línia
  - f --> mostra a temps real les línies que s'introdueixen en un fitxer

# Comandes

- cat --> mostra contingut
- awk --> té llenguatge propi

# Comandes

- sed i expressions regulars:

- echo "hola 1 vaya adios 3" | sed 's/hola \([0-9]\).\* adios \([0-9]\)/\1 \2/g'

¿?

# Operacions

- ';' separació de comandes
  - comanda1 ; comanda2 ; comanda3
- '&&' --> AND
  - comanda1 && comanda2 && comanda3
    - Si comanda1 retorna error (p.e. 1) no es continúa la execució de comandes
- '||' --> OR
  - comanda1 || comanda2 || comanda3
    - Continúa execució de comandes mentres que no s'executin correctament

# Operacions

- Operacions aritmètiques
  - expr --> compte amb caràcters especials (\\*)
    - » expr 11 \\* 2
      - 22
  - bc --> treballa amb presició arbitrària
    - Conversor pts - euros:
      - » echo \$var '/ 166.386' | bc -l
- Aquestes són comandes externes, millor fer ús de les pròpies de la shell -->



# Operacions

- `id++ id--` primer s'utilitza valor
- `++id --id` primer s'incrementa o decrementa
- `- +` menys, més
- `! ~` negació lògica i de bits
- `**` exponencial
- `* / %` multiplicació, divisió, reste
- `<< >>` shift esquerra i shift dreta
- `<= >= < >` comparacions
- `== !=` igualtat, desigualtat
- `&` bit AND
- `^` bit XOR
- `|` bit OR
- `&&` logical AND
- `||` logical OR
- `expressió1?expressió2:expressió3` evaluació condicional: si expressió1 llavors expressió2, sinó expressió3
- `= *= /= %= += -= <<= >>= &= ^= |=` assignacions

# Operacions

- Les variables de la shell per defecte tenen format caràcter
- Definir una variable de tipus numèric:
  - » **typeset -i** variable
- Avaluar:
  - » **let +=k**

# Operacions

- ¿Qué mostraria el següent script?
  - typeset -i j=7
  - typeset -i k
  - typeset -i m
  - let j=j+3
  - let j+=3
  - let k=j%3
  - let m=j/3
  - echo '(' \$m '\*3 ) +' \$k '=' \$j

¿?

# Bash 2

Administració de Sistemes Operatius  
Dept. Matemàtica Aplicada i Anàlisi,  
Universitat de Barcelona

# Index

- Estructures de control
- Altres comandes
- Útils
- Exercicis

# Estructures de control

- if condició ; then comandes ; fi
- if condició ; then comandes ; else comandes ; fi
- if condició ; then comandes ; elif condició ; then comandes ; fi
- ';' es pot substituir per enter

# Estructures de control

- while condició ; do comandes ; done
- for variable in vector ; do comandes ; done
  - **for i in \$(seq 1 10) ; do echo \$i ; done**
    - --> 1 2 3 4 5 6 7 8 9 10
    - \$(seq 1 10) ó `seq 1 10` (accent obert) és el resultat d'aplicar la comanda
    - **for (( var=1; var<=10; var++ ))**
- until condició ; do comandes ; done
- select variable in un dos tres quatre ; do ; comandes ; done
  - --> mostra menú per inserir una de les possibilitats de la llista del select

# Estructures de control

- case variable in  
patró1)  
comandes1  
;;  
**patró2 | patró3)**  
comandes2  
;;  
patróN)  
comandesN  
;;  
\*)  
comandes\*  
;;  
esac

--> Entra en cas de no coincidència prèvia



# Estructures de control-Condicions

- Condició : [espai ... X ... espai] ó test “condició”
  - X=
    - » ! (not)
    - » -a (and)
    - » -o (or)
    - » != (diferència de cadenes)
    - » = (per igualació de cadenes)
    - » -e (¿existeix fitxer?)
    - » -d (¿existeix i és un directori?)
    - » -f (¿existeix i és fitxer corrent?)
    - » -L (¿existeix i és enllaç simbòlic?)
    - » -r (¿existeix fitxer i es pot llegir?)
    - » -w (¿existeix fitxer i es pot escriure?)
    - » -x (¿existeix fitxer i es pot executar?)
    - » -z (¿existeix fitxer amd tamany = 0?)
    - » -s (¿existeix fitxer amd tamany > 0?)
    - » fitxer1 -nt fitxer2 (true si primer més actual)
    - » fitxer1 -ot fitxer2 (true si segon més actual)
    - » fitxer1 -nt fitxer2 (true si tenen mateix número de device i inode)
    - » -n (true si longitud cadena > 0)

# Estructures de control-Condicions

- Condició : [espai ... X ... espai] ó test “condició”
  - X=
    - » (exp) per evaluació d'expressions lògiques
    - » -c (¿existeix i és un dispositiu especial de caràcters?)
    - » -b (¿existeix i és un dispositiu especial de blocs?)
    - » -p (¿existeix i és un pipe?)
    - » -u (¿existeix i té bit set-user-ID?)
  - números:
    - eq equal
    - gt greater than
    - ge greater or equal
    - lt less than
    - le less or equal
    - ne not equal
- man test --> possibilitats per a les condicions

# Estructures de control-Condicions

- `if [ ! -f hola ] ; then echo "no existeix arxiu regular" ; fi -->` mostrar missatge si no existeix l'arxiu 'hola' i es de tipus regular
- Operacions sobre la condició:
  - »! condició1
  - »condició1 || condició2 || condició3 ...
  - Amb la primer TRUE valida
    - »condició1 && condició2 && condició3 ...
  - Amb el primer FALSE atura i no valida

# Estructures de control-break,continue

- break --> provoca sortida del bucle en el que es troba
- continue --> fa un salt a la següent iteració del bucle on es troba
  - » continue 4 --> es salta 4 iteracions del bucle on es troba

# Estructures de control-Conditions

- `test ! -w fichero && echo "Cuidado fichero sin permiso de escritura"`
- `test -w fichero || echo "Cuidado fichero sin permiso de escritura"`

# Comandes

- sort --> ordenar
- paste --> juntar fitxers
- nl --> numera les línies al fitxer
- tee --> redirecció de la sortida a múltiples fitxers
- join --> comparar cadenes de text a fitxers
- uniq --> comparar línies
- diff --> diferències entre fitxers
- cmp --> comparació de fitxers (mostra primera línia diferent)

# Útils

- **alias** --> equivalència de comandes: alias dir = 'ls -la'
- **unalias** dir (eliminar l'equivalència 'dir')
- **whereis** comanda --> directori on es troba la comanda
- **date '+%D %T'** --> data y hora
- **set | less** --> llista totes les variables i funcions definides
- **time** comanda:
  - En cas de dubtes entre diferents possibilitats, la comanda time retorna el temps d'execució d'una determinada comanda

# Útils

```
Bucle amb let (comanda interna)
```

```
typeset -i N=0
```

```
time while [$N -le 20000]
```

```
do let N+=1
```

```
done
```

```
real 0m1.413s
```

```
Bucle amb expr (comanda externa)
```

```
typeset -i N=0
```

```
time while [$N -le 100]
```

```
do N=`expr $N + 1`
```

```
done
```

```
real 0m1.311s
```



# Exercicis

- 1) ¿Qué retorna el següent script?

```
typeset -i j=0
while true
do
 let j=j+1
 if [$j -et 3]
 Then continue
 fi
 if [$j -et 4]
 Then continue
 fi
 if [$j -lt 10]
 then echo `expr $j * 2`
 else break
 fi
done
```

# Exercicis

- 1) ¿Qué retorna el següent script?

```
typeset -i j=0
while true
do
 let j=j+1
 if [$j -et 3]
 Then continue
 fi
 if [$j -et 4]
 Then continue
 fi
 if [$j -lt 10]
 then echo `expr $j * 2`
 else break
 fi
done
```

2  
4  
10  
12  
14  
16  
18

# Exercicis

- 2) Llistar tots els arxius d'un directori que tinguin permís de lectura per als altres
- 
- 3) Un script que llegeixi un arxiu d'entrada i guardi en un de sortida totes les línies que comencin per vocal (tant majúscula com minúscula)

# Exercicis

- 2) Llistar tots els arxius d'un directori que tinguin permís de lectura per als altres

`ls -la | grep '^.....r'`

- 3) Un script que llegeixi un arxiu d'entrada i guardi en un de sortida totes les línies que comencin per vocal (tant majúscula com minúscula)

`grep '^[AEIOUaeiou]' entrada > sortida`

# Exercicis

4) Fer un script que copii tots els arxius executables del directori actual al directori bin de l'usuari, mostri els noms dels arxius que mou i indiqui la quantitat. Si el directori de destí no existeix s'ha de crear.

# Exercicis

4) Fer un script que copii tots els arxius executables del directori actual al directori bin de l'usuari, mostri els noms dels arxius que mou i indiqui la quantitat. Si el directori de destí no existeix s'ha de crear.

```
If [! -d $HOME/bin]
then
 mkdir $HOME/bin
fi
N=0
for ARCH in *
do
 if [-x $ARCH -a -f $ARCH]
 then
 cp $ARCH $HOME/bin
 N=`expr $N + 1`
 fi
done
echo "S'han copiat $N arxius"
```

# Exercicis

5) Fer un script que mostri el major de tres números passats com a paràmetre. Tenir en compte com a condició d'error: número correcte de paràmetres.

6) Fer un 'case' per avaluar operacions d'entrada de tipus: número operació número, on operació pot ésser: +, -, x, /. x ó X s'utilitzaran per multiplicació. Tenir en compte com a condició d'error: necessari 3 paràmetres d'entrada i operació vàlida.

7) Fer un script que sumi els dígitos d'un número: 123  
 $1+2+3=6$

# Exercicis (5)

```
if [$# -ne 3]
then
 echo "Número incorrecte de paràmetres"
 exit
fi
n1=$1 ; n2=$2 ; n3=$3
if [$n1 -gt $n2] && [$n1 -gt $n3]
then
 echo "n1 és més gran"
elif [$n2 -gt $n1] && [$n2 -gt $n3]
then
 echo "n2 és més gran"
elif [$n3 -gt $n1] && [$n3 -gt $n2]
then
 echo "n3 és més gran"
elif [$n1 -eq $n2] && [$n1 -eq $n3] && [$n2 -eq $n3]
then
 echo "Els tres números són iguals"
else
 echo "hi han errors..."
fi
```



# Exercicis (6)

```
if test $# = 3
then
 case $2 in
 +) let z=$1+$3;;
 -) let z=$1-$3;;
 /) let z=$1/$3;;
 x|X) let z=$1*$3;;
 *) echo "Operació no vàlida"
 esac
 echo "Solució: $z"
else
 echo "Número de paràmetres incorrecte"
fi
```

# Exercicis

7) Fer un script que sumi els dígitos d'un número: 123

$$1+2+3=6$$

```
n=$1
```

```
sum=0
```

```
sd=0
```

```
while [$n -gt 0]
```

```
do
```

```
 sd=`expr $n % 10`
```

```
 sum=`expr $sum + $sd`
```

```
 n=`expr $n / 10`
```

```
done
```

```
 echo "La suma és $sum"
```

# Exercicis

8) Fer un script per a cada una de les següents seqüències:

```
1
12
123
1234
12345
```

A

```
|_
|_|
|_|_|
|_|_|_|
|_|_|_|_|
```

B

```
*
* *
* * *
* * * *
* * * * *
* * * * *
* * * *
* * *
* *
*
```

C

```
|Linux_____
|Linux|Linux_____
|Linux|Linux|Linux_____
|Linux|Linux|Linux_____>> Powerd Server.
|Linux|Linux~~~~~
|Linux~~~~~
```

D

```
 .
 . .
 . . .

.
```

E

```
 .
 . .
 . . .

.
```

F

# Exercicis

8) Fer un script per a cada una de les següents seqüències:

(A)

```
1
12
123
1234
12345
```

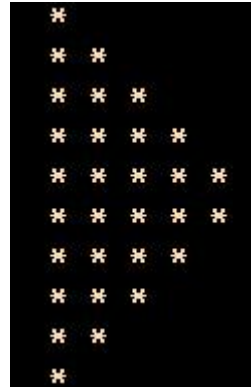
```
for ((i=1; i<=5; i++))
do
 for ((j=1; j<=i; j++))
 do
 echo -n "$j"
 done
 echo ""
done
```

# Exercicis

8) Fer un script per a cada una de les següents seqüències:

```
for ((i=1; i<=5; i++))
do
 for ((j=1; j<=i; j++))
 do
 echo -n "*"
 done
 echo ""
done
```

(C)



```
done
for ((i=5; i>=1; i--))
do
 for ((j=1; j<=i; j++))
 do
 echo -n "*"
 done
 echo ""
done
```

done

# Bash - AWK

Administració de Sistemes Operatius  
Dept. Matemàtica Aplicada i Anàlisi,  
Universitat de Barcelona

# Index

- Background
- Estructura i execució de AWK
- Caràcters especials
- Funcions, comandes i variables
- Estructures de control
- Exercicis

# Background

- Des de el seu naixement al 1978 s'ha convertit en un llenguatge molt potent i versàtil
- La paraula AWK s'usa tant per a fer referència a un llenguatge de manipulació de fitxers de dades com per referir-se al seu intèrpret
- Molt útil per processar fitxers de dades en format de text i taules
- Permet entre altres coses conversió de formats entre sortida i entrada de diferents programes



# Estructura de AWK

- BEGIN { ... }: s'executa abans de processar les línies d'entrada
- { Pot tenir una part central que es processa sencera per cada línia d'entrada i pot tenir diferents blocs '{ }'. En cas d'expressió regular s'executarà quan l'entrada es satisfaci: intruccions ó condició-instruccions }
- END { ... }: Es processarà quan acabi tota la resta

# Estructura de AWK

```
BEGIN { print "Hola" }
{ print "... y ..." }
END { print "adiós."}
```

```
debian:~/problemas/AWK# ls -la
total 3
drwxr-xr-x 2 root root 1024 2007-03-20 13:53 .
drwxr-xr-t 3 root estudiantes 1024 2007-03-20 13:50 ..
-rw-r--r-- 1 root root 69 2007-03-20 13:53 awk1
```

```
debian:~/problemas/AWK# ls -la | awk -f awk1
Hola
... y ...
... y ...
... y ...
... y ...
adiós.
```

- Cada línia del cos de AWK s'executa tants cops com línies d'entrada
- BEGIN i END en cas d'existir només a l'inici i final
- Podem fer ús de les mateixes expressions regulars que vam estudiar

# Estructura de AWK

Execució:

```
awk 'programa' fitxer1 fitxer2 ... fitxerN
```

```
awk 'programa'
```

```
awk -f programa_fitxer fitxer1 fitxer2 ... fitxerN
```

# Caràcters especials

- # serveix per comentar línies de codi
- \a Produeix un petit soroll a la terminal
- \b Retrocés
- \f Salt de pàgina
- \n Salt de línia
- \r Retorn de carro
- \t Tabulador horitzontal
- \v Tabulador vertical
- \ddd Caràcter representat en octal per 'ddd'
- \xhex Caràcter representat en hexadecimal per 'hex'
- \c Caràcter 'c'

# Caràcters especials

- El separador per defecte és un o més espais o tabulacions consecutives: `[\ \t]+`
- Per indicar separador incloure FS a BEGIN: `BEGIN { FS = "separador" }`
- `$1, $2, ...` : camp 1, camp2, ...
- `$0` : registre complet
- `NR` : número de registre
- `RS` : separador de registre (defecte: `\n`)
- `OFS` : separador de camps a la sortida (defecte: `res`)
- `ORS` : separador entre registres analitzats del cos `awk` a la sortida (`\n`)
- `NF` : número de camps
- `FNR` : número de registres llegits del fitxer actual

# Caràcters especials

- SUBSEP : separador d'índex d'array
- IGNORECASE : si diferent de 0 les expressions regulars no distingeixen entre minúscules i majúscules
- ARGV, ARGV --> com a C
  - awk 'program' arxiu
    - ARGV=3
    - ARGV[0]="awk"
    - ARGV[1]="program"
    - ARGV[2]="arxiu"
- ENVIRON : array que conté índex a les variables de retorn
- FILENAME : fitxer actual de lectura
- RLENGTH : longitud de la cadena trobada amb funció 'match'
- RSTART : índex d'inici de la cadena trobada amb 'match'



# Caràcters especials

- Exemple:

- Fitxer:

```
A | B | 10
```

```
C | D | 20
```

- Per AWK:

```
BEGIN { FS = "\\|" }
```

```
{ print NR, " ", "$1", " ", "$2", " ", "$3" }
```

- Sortida

```
1, A , B , 10
```

```
2, C , D , 20
```

- Per fixar FS com a paràmetre:

```
awk -Fseparador `programa` fitxer
```

# Caràcters especials

- **Patrons:**
- Exemple amb condicions de **patrons**
  - // conté el patró de búsqueda
  - Mostra totes les línies que continguin **patró**
  - `awk '/patró/ {print $0}' arxiu`
- **Separadors**
- Es pot separar una comanda en més d'una línia amb un enter després de:  
, { ? : || && do else  
O en qualsevol lloc amb `\  
• De la mateixa manera per haver-hi més d'una sentència per línia separant-les amb `;



# Caràcters especials

- Accés a registres
- Operacions per accedir a registres  
awk '{print \$(2\*2)}' arxiu
- Modificant els registres  
awk '{\$3 = \$2 - 10; print \$2, \$3}' arxiu
- Podem incloure un nou camp fora del rang i \$0 s'actualitza
- Accedir a un element inexistent retorna una cadena buida

# Caràcters especials

- Es pot fer redirecció (<.>, >>, <<)
- **Expressions condicionals**  
Segueixen el format de C:  
 $x > 0 ? x : -x$   
 $x == y ? a[i++] : b[i++]$

# Funcions, comandes i variables

- Printf: similar a C : es pot usar un número y un signe per definir l'espai reservat per cada variable i la seva alineació.
  - » `echo | awk '{ printf "#%-10d#%-10s#\n", 77, "mundo" }'`
  - » `#77          #mundo          #`

| Especificación de formato de datos para 'printf' |                                                   |
|--------------------------------------------------|---------------------------------------------------|
| %c                                               | Carácter ASCII                                    |
| %d                                               | Entero representado en decimal                    |
| %e                                               | Coma flotante (exponente = e[+-]dd)               |
| %E                                               | Coma flotante (exponente = E[+-]dd)               |
| %f                                               | Coma flotante sin exponente                       |
| %g                                               | Equivale al más corto de los formatos 'e' o 'f'   |
| %G                                               | Equivale al más corto de los formatos 'E' o 'F'   |
| %o                                               | Entero representado en octal                      |
| %s                                               | Cadena de caracteres                              |
| %x                                               | Entero representado en hexadecimal con minúsculas |
| %X                                               | Entero representado en hexadecimal con mayúsculas |
| %%                                               | Carácter '%'                                      |

# Funcions, comandes i variables

- Variables i operacions

| Operadores aritméticos    |                                                                   |
|---------------------------|-------------------------------------------------------------------|
| +                         | Suma                                                              |
| -                         | Resta                                                             |
| *                         | Multiplicación                                                    |
| /                         | División                                                          |
| %                         | Módulo (resto)                                                    |
| ^                         | Potenciación                                                      |
| Operadores de asignación. |                                                                   |
| var = expr                | Asignación                                                        |
| var ++                    | Incrementa la variable en una unidad                              |
| var --                    | Decrementa la variable en una unidad                              |
| var += expr_aritm         | Incrementa la variable en cierta cantidad                         |
| var -= expr_aritm         | Decrementa la variable en cierta cantidad                         |
| var *= expr_aritm         | Multiplica la variable por cierta cantidad                        |
| var /= expr_aritm         | Divide la variable por cierta cantidad                            |
| var %= expr_aritm         | Guarda en la variable el resto de su división por cierta cantidad |
| var ^= expr_aritm         | Eleva el valor de la variable en cierta cantidad                  |

# Funcions, comandes i variables

- Variables i operacions

| Operadores lógicos y de relación.                   |                             |
|-----------------------------------------------------|-----------------------------|
| <code>expr_aritm ==<br/>expr_aritm</code>           | Comparación de igualdad     |
| <code>expr_aritm !=<br/>expr_aritm</code>           | Comparación de desigualdad  |
| <code>expr_aritm &lt; expr_aritm</code>             | Comparación menor que       |
| <code>expr_aritm &gt; expr_aritm</code>             | Comparación mayor que       |
| <code>expr_aritm &lt;=<br/>expr_aritm</code>        | Comparación menor igual que |
| <code>expr_aritm &gt;=<br/>expr_aritm</code>        | Comparación mayor igual que |
| <code>expr_cad ~<br/>expr_regular</code>            | Se ajusta al patrón         |
| <code>expr_cad !~<br/>expr_regular</code>           | No se ajusta al patrón      |
| <code>expr_logica   <br/>expr_logica</code>         | Operador lógico OR (O)      |
| <code>expr_logica &amp;&amp;<br/>expr_logica</code> | Operador lógico AND (Y)     |
| <code>! expr_logica</code>                          | Operador lógico NOT (NO)    |

# Funcions, comandes i variables

- funcions

| Funciones aritméticas. |                                                             |
|------------------------|-------------------------------------------------------------|
| atan2( y, x)           | Retorna el arco-tangente de y/x en radianes                 |
| cos(x)                 | Retorna el coseno de x en radianes                          |
| exp(x)                 | Retorna el exponencial de x ( $e^x$ )                       |
| int(x)                 | Retorna el valor entero de x truncado la parte decimal      |
| log(x)                 | Retorna el logaritmo neperiano de x                         |
| rand()                 | Retorna un valor seudo aleatorio comprendido entre 0 y 1    |
| sin(x)                 | Retorna el seno de x en radianes                            |
| sqrt(x)                | Retorna la raiz cuadrada de x                               |
| srand(x)               | Inicializa la semilla para generar números pseudoaleatorios |

# Funcions, comandes i variables

- funcions

| Funciones para usar con cadenas de caracteres |                                                                                                                                                                                                                 |
|-----------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| gsub(r, s, t)                                 | Sustituye 's' globalmente en todo 't' cada vez que se encuentre un patrón ajustado a la expresión regular 'r'. Si no se proporciona 't' se toma \$0 por defecto. Devuelve el número de sustituciones realizado. |
| index(cadena, subcadena)                      | Retorna la posición de la 'subcadena' en 'cadena' (Primera posición = 1)                                                                                                                                        |
| length(cadena)                                | Devuelve la longitud de la 'cadena'. Tomará \$0 por defecto si no se proporciona 'cadena'                                                                                                                       |
| split(cadena, array, sep)                     | Parte 'cadena' en elementos de 'array' utilizando 'sep' como separador. Si no se proporciona 'sep' se usará FS. Devuelve el número de elementos del array                                                       |
| sub(r, s, t)                                  | Sustituye 's' en 't' la primera vez que se encuentre un patrón ajustado a la expresión regular 'r'. Si no se proporciona 't' se toma \$0 por defecto. Devuelve 1 si tiene éxito y 0 si falla                    |
| substr(cadena, beg, len)                      | Devuelve una subcadena de 'cadena' que empieza en 'beg' con una longitud 'len'. Si no se proporciona longitud devuelve hasta el final de la cadena desde 'beg'                                                  |
| tolower(cadena)                               | Pasa a minúsculas                                                                                                                                                                                               |
| toupper(cadena)                               | Pasa a mayúsculas                                                                                                                                                                                               |

# Funcions, comandes i variables

- funcions

| Algunas otras funciones         |                                                                     |
|---------------------------------|---------------------------------------------------------------------|
| match(cadena, expr_reg)         | Indica si 'cadena' se ajusta o no a la expresión regular 'expr_reg' |
| system(comando)                 |                                                                     |
| sprintf(formato [, expr-list] ) | Para obtener salida con formato.                                    |



# Funcions, comandes i variables

- Exemple: Càlcul amb registres

1/1/2007|AGUA|100

2/2/2007|LUZ|200

3/3/2007|LUZ|300

---

```
BEGIN {
 FS="\\|";
 cont_reg_luz=0;
 cont_importe_luz=0;
}
$2 ~ /LUZ/ {
 cont_reg_luz = cont_reg_luz + 1;
 cont_importe_luz = cont_importe_luz + $3 ;
 printf "%3d,%3d,%s,%s,%s,%10d\\n", NR, cont_reg_luz, $1, $2, $3, cont_importe_luz;
}
END {
 printf("Consumo promedio = %d\\n", cont_importe_luz / cont_reg_luz);
}
```

---

2, 1, 2/2/2007, LUZ, 200, 200

3, 2, 3/3/2007, LUZ, 300, 500

Consumo promedio = 250

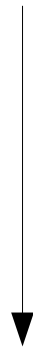
# Funcions, comandes i variables

- Pas de paràmetres a awk:

awk **-v variable1=valor -f arxiu variable2=valor2 variable3=valor3**



per accedir-hi desde BEGIN



Ús directe del nom de la variable dins del cos de les funcions de  
awk

# Funcions, comandes i variables

- Declaració de funcions

```
function nom_funció (paràmetres) {
sentències
}
```

comandes AWK i crida de funcions...

--> podem fer **return valor** sense indicar tipus a la funció

- Funció system

```
system("ls")
```

```
system("test -f" nom_fitxer)
```

# Funcions, comandes i variables

- Funció **getline**
- És una funció que s'utilitza com una sentència
- Retorna 1 si llegeix una línia, 0 si arriba al final d'entrada, i -1 si es produeix un error
- Getline directament assigna l'entrada a \$0 y els camps a \$1, \$2, \$3, etc.
- Per no alterar el valor de \$0 podem assignar el valor de getline a una variable: "getline variable"
- Redirecció per entrada de fitxer "getline < fitxer"
- També amb pipe "comanda | getline variable"
- Getline té moltes més possibilitats.
- Cal anar amb compte de tancar les comandes obertes amb pipes i fitxers oberts: close(comanda), close(fitxer)

# Funcions, comandes i variables

- **Arrays associatius**
- Podem utilitzar-los com en bash:
- `array[posició]=valor`, utilitzar `printf` per mostrar continguts
- No és necessari definir la longitud del vector ni definir ordre entre els seus elements
- Es crea dinàmicament i associació per Hashing
- Índex amb text:
  - `array["text1"]=valor`

# Funcions, comandes i variables

- **Arrays associatius**

Element "dog" valor "chien"

Element "cat" valor "chat"

Element "one" valor "un"

Element 1 valor "un"

- Podem saber si existeix un element a un array amb:
  - **índex in array**
- Recorre tots els elements d'un array:
  - for ( variable in array )
- Esborrar una posició de l'array
  - delete array[índex]

# Funcions, comandes i variables

- array multidimensional  
array[x,y]
- Un array n-dimensional concatena els n índex mitjançant **SUBSEP**:
  - array[x,y] == array[x SUBSEP y]
  - Per defecte SUBSEP = “\034” caràcter ASCII poc freqüent
- Evitar que la concatenació de diferents posicions sigui la mateixa
- Per buscar si existeix una seqüència de índexs en un array:  
(subindex1, ... , subindexN) in array

# Funcions, comandes i variables

- **array multidimensional**
- A la pràctica els arrays multidimensionals es tracten com 1-dimensionals
- Per recorre un array n-dimensional podem fer:

```
for (índex in array) {
 split(índex, array_index, SUBSEP)
```

```
 ...
```

```
}
```

- On a array\_index es guarden les n-posicions



# Estructuras de control

|                 |                                       |
|-----------------|---------------------------------------|
| <b>if</b>       | if (condicion) { accion }<br>else { } |
| <b>while</b>    | while (condicion) { accion }          |
| <b>do</b>       | do { accion } while (condicion)       |
| <b>break</b>    | Para salir de un ciclo o loop         |
| <b>continue</b> | Para ir al proximo ciclo              |
| <b>exit</b>     | Sale                                  |
| <b>next</b>     | Para a la siguiente linea             |



# Exercicis

- 1) Donat el següent fitxer `expreg.awk`:

```
BEGIN { print "Érase una vez..." }
/^$/ { print "Línea vacía" }
/[0-9]+/ { print "Tiene un número" }
/\.$/ { print "Termina con punto" }
Comentario
{ print "-----" }
END { print "... y se acabó." }
```

- Y el fitxer de dades `expreg.dat`:

Línea número 1.

Línea número 2

...

Fin de los datos

- ¿qué mostraria `awk -f expreg.awk expreg.dat` ?

# Exercicis

- 2) Mostrar el total de línies d'un fitxer que continguin la paraula 'exercici'
- 3) Mostrar el valor absolut dels registres negatius d'un fitxer (a la sortida per pantalla)
- 4) fer una comanda awk per mostrar les línies d'un fitxer amb longitud més gran que 60

# Exercicis

- 5) fer un script awk per imprimir un fitxer d'entrada amb tots els registres en ordre invers:  
registre1 , ... , registreN --> registreN , ... , registre1
- 6) Fer un programa awk que mostri totes les paraules diferents que hi ha a un fitxer junt amb la quantitat d'ocurrències de cadascuna

# Bash – AWK exercicis resolts

Administració de Sistemes Operatius  
Dept. Matemàtica Aplicada i Anàlisi,  
Universitat de Barcelona

# Exercici 1

Érase una vez...

Tiene un número

Termina con punto

-----

Tiene un número

-----

Termina con punto

-----

Línea vacía

-----

-----

... y se acabó.

# Exercicis

2)

```
/paraula/ {nlines = nlines +1}
```

```
END {print nlines}
```

3)

```
{ for (i = 1; i < NF; i=i+1) if ($i < 0) { $i=-$i ; print $i}
```

# Exercicis

4) `awk 'length($0) > 60' arxiu`

5) `{for (I=NF ; I>0; --I) print $I}`

6)

{

for (i = 1; i<= NF; i++)

freq[\$i]++

}

END {

for (word in freq)

printf "%s \t%d\n", word, freq[word]

}



# Python

Administració de Sistemes Operatius  
Dept. Matemàtica Aplicada i Anàlisi,  
Universitat de Barcelona

# Índex I

- Introducció/característiques
- Estructura
- Assignacions
- Operacions
- Llibreries
- Strings
- Estructures de control i excepcions
- Fitxers
- Objectes i classes
- Col·leccions

# Introducció/Característiques

- Primera versió de 1990
- Simple (minimalista)
- Alt nivell
- Codi obert (portable)
- Interpretat
- Tipus dinàmic
- Orientat objectes
- Incrustable (C++, Java, etc.)
- Comparat amb Perl

# Estructura

Si cridem a l'interpret de Python apareix:

```
Python 2.5.1(r251:54863, Apr 18 2007, 08:51:08) [MSCv.1310 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
```

```

Personal firewall software may warn about the connection IDLE
makes to its subprocess using this computer's internal loopback
interface. This connection is not visible on any external
interface and no data is sent to or received from the Internet.

```

```
IDLE 1.2.1
>>>
```

i ja podem començar a donar ordres...

```
>>> print "Hello, World"
Hello, World
>>> print 2 + 3
5
>>> print "2 + 3 =", 2 + 3
2 + 3 = 5
```

# Estructura

Si volem executar una seqüència d'instruccions podem crear/**definir** una **funció** (que en aquest cas s'anomena `hello`):

```
>>> def hello():
 print "Hello"
 print "Computers are Fun"
>>>
```

Identació per indicar  
que són part de la  
definició de la funció

Línia en blanc per indicar que és el final de la  
definició

Un cop la tenim definida la poden **cridar/invocar**:

```
>>> hello()
Hello
Computers are Fun
>>>
```

# Estructura

Hi ha diverses maneres de cridar el programa, però la més bàsica és aquesta:

```
>>> import chaos
This program illustrates a chaotic function
Enter a number between 0 and 1: .25
0.73125
0.76644140625
0.698135010439
0.82189581879
0.570894019197
0.955398748364
0.166186721954
0.540417912062
0.9686289303
0.118509010176
>>>
```

Aquesta línia indica a l'interpret que volem carregar el mòdul `chaos` del fitxer `chaos.py` i a continuació l'executa.

Quan importem un mòdul d'aquesta manera, l'interpret crea un fitxer intermedi anomenat `chaos.pyc` (i que conté el *byte code*) que fa servir en el procés d'interpretació/compilació. Els fitxers `.pyc` poden servir per anar més ràpid en el futur...

El mòdul resta carregat durant tota la **sessió**, i podem cridar-lo amb la instrucció `chaos.main()` Si no haguéssim creat la funció `main` no podríem!

Si la volem recarregar (per exemple, perquè l'hem editat en paral·lel) : `reload(chaos)`

Python té una llista de llocs on espera trobar la funció (començant pel directori des d'on l'hem cridat). Per fer-ho des d'un altre lloc se li ha de dir! (`import sys+ syspath`)

# Estructura

Línies de comentari

```
File: chaos.py
A simple program illustrating chaotic behavior.
```

```
def main():
```

Els programes es solen posar en una funció anomenada main.

```
 print "This program illustrates a chaotic function"
```

$x$  és una **variable** i serveix per donar nom a un valor i així poder cridar-lo quan ens interessi.

```
 x = input("Enter a number between 0 and 1: ")
```

Això és una instrucció `input`, i el que fa és escriure-ho i esperar una resposta acabada per `<Enter>`

```
 for i in range(10):
```

Això és una instrucció de tipus `iteració`.

```
 x = 3.9 * x * (1 - x)
```

Aquest és el cos de la instrucció `iteració`. La primera instrucció és una **assignació**.

```
 print x
```

```
main()
```

Aquesta línia crida la funció.

# Assignacions

- **Assignacions**

**Assignacions simultànies:**

```
<var>, <var>, ..., <var> = <expr>, <expr>, ..., <expr>
```

com per exemple

```
sum, diff = x+y, x-y
```

Aquest tipus d'assignació pot ser molt útil, com per exemple per intercanviar els valors de dues variables. Això no funciona!:

```
x = y
y = x
```

```
variables x y
initial values 2 4
x = y
now 4 4
y = x
final 4 4
```



# Operacions

| operator | operation      |
|----------|----------------|
| +        | addition       |
| -        | subtraction    |
| *        | multiplication |
| /        | division       |
| **       | exponentiation |
| %        | remainder      |
| abs ()   | absolute value |

: Python built-in numeric operations.

```
>>> 3.0 + 4.0
7.0
>>> 3 + 4
7
>>> 3.0 * 4.0
12.0
>>> 3 * 4
12
>>> 10.0 / 3.0
3.333333333333333
>>> 10 / 3
3
>>> 10 % 3
1
>>> abs(5)
5
>>> abs(-3.5)
3.5
```

# Llibreries

Python també ens dona funcions matemàtiques dins d'una **biblioteca** (*library*) especial. Una biblioteca no és res més que un mòdul que conté definicions útils.

```
quadratic.py
A program that computes the real roots of a quadratic equation.
Illustrates use of the math library.
Note: this program crashes if the equation has no real roots.

import math # Makes the math library available.

def main():
 print "This program finds the real solutions to a quadratic"
 print

 a, b, c = input("Please enter the coefficients (a, b, c): ")

 discRoot = math.sqrt(b * b - 4 * a * c)
 root1 = (-b + discRoot) / (2 * a)
 root2 = (-b - discRoot) / (2 * a)

 print
 print "The solutions are:", root1, root2

main()
```

$$ax^2 + bx + c = 0$$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

# Strings

Strings:

```
>>> firstName = input("Please enter your name: ")
Please enter your name: "John"
>>> print "Hello", firstName
Hello John
```

O així:

```
>>> firstName = raw_input("Please enter your name: ")
Please enter your name: John
>>> print "Hello", firstName
Hello John
```

Que és el mateix que `input ( )`, però sense avaluar l'expressió que entrem.

# Strings

Per accedir als elements d'un *string* hem de veure com Python els indexa:

|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
| H | e | l | l | o |   | B | o | b |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

Llavors podem accedir als valors de cada element de la seqüència o fins i tot a subseqüències:

```
>>> greet = "Hello Bob"
>>> greet[0]
'H'
>>> print greet[0], greet[2], greet[4]
H l o
>>> x = 8
>>> print greet[x-2]
B

>>> greet[0:3]
'Hel'
>>> greet[5:9]
' Bob'
>>> greet[:5]
'Hello'
>>> greet[5:]
' Bob'
>>> greet[:]
'Hello Bob'
```

# Strings

```
>>> import string
>>> string.split("Hello string library!")
['Hello', 'string', 'library!']
```

Dividim strings en troços (blancs)

```
>>> string.split("32,24,25,57", ",")
['32', '24', '25', '57']
```

També podem especificar per "on" dividir!

```
>>> eval("345.67")
```

```
345.67
```

```
>>> eval("3+4")
```

```
7
```

```
>>> x = 3.5
```

```
>>> y = 4.7
```

```
>>> eval("x * y")
```

```
16.45
```

```
>>> x = eval(raw_input("Enter a number "))
```

```
Enter a number 3.14
```

```
>>> print x
```

```
3.14
```

```
>>> string.split("87 104 97 116 32 97 32 83 111 117 114 112 117 115 115 33")
['87', '104', '97', '116', '32', '97', '32', '83', '111', '117', '114', '112', '117', '115', '115', '33']
```

# Estructures de control i excepcions

```
if <condition>:
 <body>
```

```
averagel.py
```

```
def main():
 n = input("How many numbers do you have? ")
 sum = 0.0
 for i in range(n):
 x = input("Enter a number >> ")
 sum = sum + x
 print "\nThe average of the numbers is", sum / n
```

```
<expr> and <expr>
```

```
<expr> or <expr>
```

- Excepcions

```
try:
 <body>
except <ErrorType>:
 <handler>
```

```
if <condition>:
 <statements>
else:
 <statements>
```

```
for <var> in <sequence>:
 <body>
```

```
while <condition>:
 <body>
```

```
i = 0
while i <= 10:
 print i
 i = i + 1
```

# Fitxers

Hi ha diversos operadors definits sobre els fitxers.

**Obrir-lo**, o assignar-li una variable que el representa

```
<filevar> = open(<name>, <mode>)

infile = open("numbers.dat", "r")
```

Lavors ja podem accedir als seus continguts

```
<filevar>.read()
<filevar>.readline()
<filevar>.readlines()

printfile.py
Prints a file to the screen.

def main():
 fname = raw_input("Enter filename: ")
 infile = open(fname, 'r')
 data = infile.read()
 print data

main()
```

Podem llegir tot el fitxer,  
tots els caràcters fins a final de línia o una seqüència  
d'strings que representen les línies del fitxer.

Si ja existia una variable  
amb aquest nom la  
perdrem!

# Fitxers

Llegir les 5 primeres línies d'un fitxer

```
infile = open(someFile, 'r')
for i in range(5):
 line = infile.readline()
 print line[:-1]
```

Processar totes les línies d'un fitxer

```
infile = open(someFile, 'r')
for line in infile.readlines():
 # process the line here
infile.close()
```



# Fitxers

Podem escriure coses al fitxer

```
<file-var>.write(<string>)
```

Obrir un fitxer per escriptura  
borra tot el seu contingut!

Per exemple, dues noves línies

```
outfile = open("example.out", 'w')
count = 1
outfile.write("This is the first line\n")
count = count + 1
outfile.write("This is line number %d" % (count))
outfile.close()
```



```
This is the first line
This is line number 2
```

# Fitxers

```
userfile.py
Program to create a file of usernames in batch mode.

import string

def main():
 print "This program creates a file of usernames from a"
 print "file of names."

 # get the file names
 inFileName = raw_input("What file are the names in? ")
 outFileName = raw_input("What file should the usernames go in? ")

 # open the files
 inFile = open(inFileName, 'r')
 outFile = open(outFileName, 'w')

 # process each line of the input file
 for line in inFile.readlines():
 # get the first and last names from line
 first, last = string.split(line)
 # create the username
 uname = string.lower(first[0]+last[:7])
 # write it to the output file
 outFile.write(uname+'\n')

 # close both files
 inFile.close()
 outFile.close()

 print "Usernames have been written to", outFileName

main()
```

# Fitxers

Quan creem un fitxer obrint-lo o escrivint-hi, el nou fitxer es col·loca al directori des d'on hem cridat el programa. Si el creem per llegir, el busca en el mateix directori.

Sinó està la mateix directori:

```
>>> f = open("/usr/share/dict/words","r")
>>> print f.readline()
Aarhus
```

Per escriure coses a un fitxer les hem de convertir a text:

```
>>> f.write (str(12.3))
>>> f.write (str([1,2,3]))
```

Però això ho fa tot molt complicat:

```
>>> f.readline()
'12.3[1, 2, 3]'
```

# Fitxers

Python ens dona una utilitat, el *pickling*, que ens ho facilita:

```
>>> import pickle
>>> f = open("test.pck", "w")

>>> pickle.dump(12.3, f)
>>> pickle.dump([1,2,3], f)
>>> f.close()

>>> f = open("test.pck", "r")
>>> x = pickle.load(f)
>>> x
12.3
>>> type(x)
<type 'float'>
>>> y = pickle.load(f)
>>> y
[1, 2, 3]
>>> type(y)
<type 'list'>
```

# Fitxers

**Iteracions en fitxers**, o com processar els elements d'un fitxer.

```
average5.py

def main():
 fileName = raw_input("What file are the numbers in? ")
 infile = open(fileName, 'r')
 sum = 0.0
 count = 0
 for line in infile.readlines():
 sum = sum + eval(line)
 count = count + 1
 print "\nThe average of the numbers is", sum / count
```

# Objectes i classes

Considerem aquest programa:

```
class Point:
 def __init__(self, x=0, y=0):
 self.x = x
 self.y = y
```

```
 def __str__(self):
 return '(' + str(self.x) + ', ' + str(self.y) + ')'
```

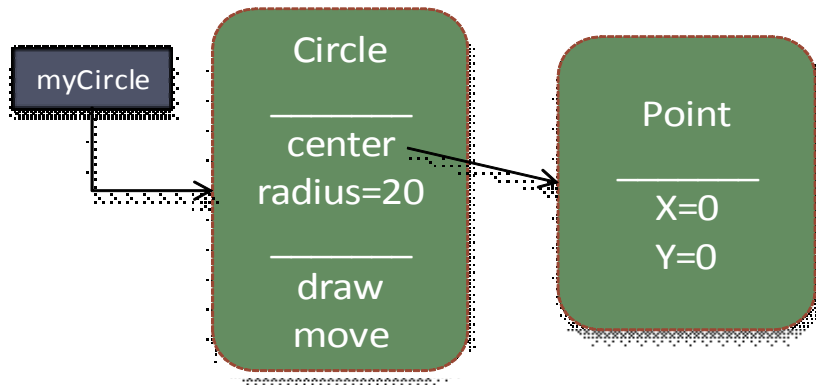
Si una classe defineix un mètode anomenat `__str__`, aquest té precedència sobre la funció `str` de Python

```
>>> p = Point(3, 4)
>>> str(p)
'(3, 4)'
```

```
>>> p = Point(3, 4)
>>> print p
(3, 4)
```

Imprimir un punt crida implícitament `__str__` de l'objecte! I per tant canvia el comportament del `print`.

# Objectes i classes



```
myCircle = Circle(Point(0,0), 20)
```

Usem el nom de la classe per invocar el constructor.

Aquesta operació crea una nova instància i emmagatzema una referència a la variable `myCircle`.

Els paràmetres que passem al constructor serveixen per a inicialitzar les variables instanciades (`center` i `radius`) dins de `myCircle`.

Un cop la instància està creada la podem manipular amb els mètodes:

```
myCircle.draw(win)
myCircle.move(dx, dy)
...
```

# Objectes i classes

```
msdie.py
Class definition for an n-sided die.

from random import randrange

class MSDie:

 def __init__(self, sides):
 self.sides = sides
 self.value = 1

 def roll(self):
 self.value = randrange(1, self.sides+1)

 def getValue(self):
 return self.value

 def setValue(self, value):
 self.value = value
```



# Col·leccions

## Exemples de col·leccions:

- Paraules d'un text.
- Estudiants d'un curs.
- Dades d'un experiment.
- Clients d'un negoci.
- Els gràfics que es poden dibuixar en una finestra.

Python ens dona suport per a la manipulació d'aquest tipus de dades.

# Col·leccions

De fet les llistes i els *strings* són conceptualment molt semblants, i podem aplicar-hi operadors semblants:

La diferència és el que contenen. Les llistes poden contenir qualsevol tipus de dades, incloent classes definides pel programador. Les llistes són mutables, és a dir, es poden canviar sobre la mateixa estructura (els *strings* no!).

| Operator            | Meaning       |
|---------------------|---------------|
| <seq> + <seq>       | Concatenation |
| <seq> * <int-expr>  | Repetition    |
| <seq>[ ]            | Indexing      |
| len(<seq>)          | Length        |
| <seq>[ : ]          | Slicing       |
| for <var> in <seq>: | Iteration     |

```
>>> myList = [34, 26, 15, 10]
>>> myList[2]
15
>>> myList[2] = 0
>>> myList
[34, 26, 0, 10]
>>> myString = "Hello World"
>>> myString[2]
'l'
>>> myString[2] = 'z'
Traceback (innermost last):
 File "<stdin>", line 1, in ?
TypeError: object doesn't support item assignment
```

# Col·leccions

| Method                                  | Meaning                                                                 |
|-----------------------------------------|-------------------------------------------------------------------------|
| <code>&lt; list &gt;.append(x)</code>   | Add element x to end of list.                                           |
| <code>&lt; list &gt;.sort()</code>      | Sort the list. A comparison function may be passed as parameter.        |
| <code>&lt; list &gt;.reverse()</code>   | Reverses the list.                                                      |
| <code>&lt; list &gt;.index(x)</code>    | Returns index of first occurrence of x.                                 |
| <code>&lt; list &gt;.insert(i,x)</code> | Insert x into list at index i. (Same as <code>list [i:i] = [x]</code> ) |
| <code>&lt; list &gt;.count(x)</code>    | Returns the number of occurrences of x in list.                         |
| <code>&lt; list &gt;.remove(x)</code>   | Deletes the first occurrence of x in list.                              |
| <code>&lt; list &gt;.pop(i)</code>      | Deletes the ith element of the list and returns its value.              |
| <code>x in &lt;list &gt;</code>         | Checks to see if x is in the list (returns a Boolean).                  |

# Col·leccions

```
>>> a = [1, 2, 3]
>>> b = [1, 2, 3]
>>> id(a)
135045528
>>> id(b)
135041704
```

```
a → [1, 2, 3]
b → [1, 2, 3]
```

Com que les variables es refereixen a objectes, si fem referir una variable a una altra tenim:

```
>>> a = [1, 2, 3]
>>> b = a
```

```
a → [1, 2, 3]
b → [1, 2, 3]
```

# Col·leccions

Com que la llista té dos noms, direm que té un **alias**:

```
>>> b[0] = 5
>>> print a
[5, 2, 3]
```

El **clonatge** és una tècnica per la que fem una còpia de l'objecte en si, no de la referència. Pel cas de les llistes ho podem fer així:

```
>>> a = [1, 2, 3]
>>> b = a[:]
>>> print b
[1, 2, 3]

>>> b[0] = 5
>>> print a
[1, 2, 3]
```

# Col·leccions

Per evitar els perills de l'alias, Python ens proporciona un mòdul de còpia:

```
>>> import copy
>>> p1 = Point()
>>> p1.x = 3
>>> p1.y = 4
>>> p2 = copy.copy(p1)
>>> p1 == p2
False
>>> samePoint(p1, p2)
True
```

# Col·leccions

Python ens proporciona un altre tipus de col·lecció no gaire estàndard, però molt útil: els **diccionaris**.

La raó de la seva existència és que no sempre serà possible accedir a una dada pel seu índex, sinó per exemple, pel seu valor (p.e. pel DNI d'un conjunt d'empleats).

És a dir, volem accedir a un valor per una **clau**.

Una col·lecció que ens permet això es diu un “*mapping*” (altres llenguatges ho anomenen *taules hash* o *vectors associatius*).

Python les crea així:

```
>>> passwd = {"guido": "superprogrammer", "turing": "genius", "bill": "monopoly"}
```

# Col·leccions

I ens permet accedir-hi així:

```
>>> passwd["guido"]
'superprogrammer'
>>> passwd["bill"]
'monopoly'
```

Els diccionaris són mutables:

```
>>> passwd["bill"] = "bluescreen"
>>> passwd
{'turing': 'genius', 'bill': 'bluescreen', 'guido': 'superprogrammer'}
```

(els diccionaris no tenen ordre, i Python els imprimeix amb un ordre propi, no el que hem entrat!)



# Col·leccions

Podem entrar-hi dades:

```
>>> passwd['newuser'] = 'ImANewbie'
>>> passwd
{'turing': 'genius', 'bill': 'bluescreen', \
 'newuser': 'ImANewbie', 'guido': 'superprogrammer'}
```

..des d'un fitxer:

```
passwd = {}
for line in open('passwords', 'r').readlines():
 user, pass = string.split(line)
 passwd[user] = pass
```

# Col·leccions

| Method                                         | Meaning                                                                     |
|------------------------------------------------|-----------------------------------------------------------------------------|
| <code>&lt;dict&gt;.has_key(&lt;key&gt;)</code> | Returns true if dictionary contains the specified key, false if it doesn't. |
| <code>&lt;dict&gt;.keys()</code>               | Returns a list of the keys.                                                 |
| <code>&lt;dict&gt;.values()</code>             | Returns a list of the values.                                               |
| <code>&lt;dict&gt;.items()</code>              | Returns a list of tuples (key, value) representing the key-value pairs.     |
| <code>del &lt;dict&gt;[&lt;key&gt;]</code>     | Delete the specified entry.                                                 |
| <code>&lt;dict&gt;.clear()</code>              | Delete all entries.                                                         |

```
>>> passwd.keys()
['turing', 'bill', 'newuser', 'guido']
>>> passwd.values()
['genius', 'bluescreen', 'ImANewbie', 'superprogrammer']
>>> passwd.items()
[('turing', 'genius'), ('bill', 'bluescreen'), ('newuser', 'ImANewbie'), \
 ('guido', 'superprogrammer')]
>>> passwd.has_key('bill')
1
>>> passwd.has_key('fred')
0
>>> passwd.clear()
>>> passwd
{}
```

# Col·leccions

Hi ha una altra classe de col·lecció a Python que és semblant a la llista, però que és immutable.: la **tupla**.

```
>>> tuple = 'a', 'b', 'c', 'd', 'e'
```

Que també es pot (i es sol) escriure com:

```
>>> tuple = ('a', 'b', 'c', 'd', 'e')
```

Si només té un element hem de posar-hi una coma, sinó crea un string!

```
>>> t1 = ('a',)
>>> type(t1)
<type 'tuple'>
```

```
>>> t2 = 'a'
>>> type(t2)
<type 'str'>
```

# Col·leccions

Les operacions són les mateixes que per les llistes:

```
>>> tuple = ('a', 'b', 'c', 'd', 'e')
>>> tuple[0]
'a'
```

```
>>> tuple[1:3]
('b', 'c')
```

```
>>> tuple[0] = 'A'
TypeError: object doesn't support item assignment
```

```
>>> tuple = ('A',) + tuple[1:]
>>> tuple
('A', 'b', 'c', 'd', 'e')
```

# Python per administrar sistemes

Administració de Sistemes Operatius  
Dept. Matemàtica Aplicada i Anàlisi,  
Universitat de Barcelona

# Índex II

- Mòduls importants
- Exemples d'administració amb Python
- Exercicis

# Mòduls importants

- `sys`:
  - Arguments de la línia de comandes
    - `sys.argv`
  - Sortir
    - `sys.exit(0)`
- `os`
  - Crear un procés
    - `os.fork()`
  - Modificar el sistema de fitxers
    - `os.mkdir("test")`
    - `os.rmdir("test")`
    - **`os.system()`**
- `re`, `per` expressions regulars, `stat`, `sys`, `os`, `string`, `commands`, `glob`, etc.

# Exemple I

```
import sys, os, string, commands
pattern = raw_input("Enter the file pattern to search for:\n")
commandString = "find " + pattern
commandOutput =
 commands.getoutput(commandString)
findResults = string.split(commandOutput, "\n")
print "Files:"
for file in findResults:
 # PROCESS EACH FILE SEPARATELY
```



# Exemple II

```
import pwd
#get user database
(passwd)
passwd_db =
 pwd.getpwall()
try:
 for entry in passwd_db:
 print entry
 # We can process
 users info using
 entry[0], etc.
```

```
('sync', 'x', 4, 65534, 'sync', '/bin', '/bin/sync')
('games', 'x', 5, 60, 'games', '/usr/games', '/bin/sh')
('man', 'x', 6, 12, 'man', '/var/cache/man', '/bin/sh')
('lp', 'x', 7, 7, 'lp', '/var/spool/lpd', '/bin/sh')
('mail', 'x', 8, 8, 'mail', '/var/mail', '/bin/sh')
('news', 'x', 9, 9, 'news', '/var/spool/news', '/bin/sh')
('uucp', 'x', 10, 10, 'uucp', '/var/spool/uucp', '/bin/sh')
('proxy', 'x', 13, 13, 'proxy', '/bin', '/bin/sh')
('www-data', 'x', 33, 33, 'www-data', '/var/www', '/bin/sh')
('backup', 'x', 34, 34, 'backup', '/var/backups', '/bin/sh')
('list', 'x', 38, 38, 'Mailing List Manager', '/var/list', '/bin/sh')
('irc', 'x', 39, 39, 'ircd', '/var/run/ircd', '/bin/sh')
('gnats', 'x', 41, 41, 'Gnats Bug-Reporting System (admin)', '/var/lib/
/bin/sh')
('nobody', 'x', 65534, 65534, 'nobody', '/nonexistent', '/bin/sh')
('Debian-exim', 'x', 100, 102, '', '/var/spool/exim4', '/bin/false')
('statd', 'x', 101, 65534, '', '/var/lib/nfs', '/bin/false')
('identd', 'x', 102, 65534, '', '/var/run/identd', '/bin/false')
('aso', 'x', 1000, 1000, 'aso,,,', '/home/aso', '/bin/bash')
('aso1', 'x', 1001, 1001, '', '/home/aso1', '/bin/bash')
('aso2', 'x', 1002, 1001, '', '/home/aso2', '/bin/bash')
('ftptest', 'x', 1003, 1002, ',,,', '/home/ftptest', '/bin/bash')
```

```
except:
 print "There was a
 problem running the
 script."
```

# Exemple II (2)

- Alternativa per llegir el fitxer:

```
import os,sys
passwd="/etc/passwd"
if not os.path.isfile(passwd):
 print "Can't open %s." % passwd
 sys.exit(1)
for line in open(passwd).readlines():
 (uname, xpass, uid, gid, junk, home_dir, junk2) =
 line.split(':')
 # process variables ...
```

# Exemple III

- Renombrar noms de fitxers:  
bridge.jpg -> bridgeB.jpg  
airplane.jpg -> airplaneB.jpg

```
#####
```

```
import glob, os
```

```
os.chdir(DIR1)
```

```
for file in glob.glob('*.*jpg'):
```

```
 (base, extension) = os.path.splitext(file)
```

```
 os.rename(file, os.path.join(DIR2, base + 'B.jpg'))
```

# Exercicis

- Exercici 1: La sèrie de Fibonacci és 1, 2, 3, 5, 8, ... Cada nombre de la seqüència (excepte els dos primers) és el resultat de la suma dels dos anteriors. Fer un script en python que calculi el nombre n-èssim de Fibonacci, sent n un paràmetre entrat per l'usuari.
- Exercici 2: Calcular  $n!/(n-m)!m!$  A partir d'una funció que rebi m i n (combinacions(m,n)), tenint en compte que n ha de ser més gran o igual que m.

# Exercicis

- Exercici 1: La sèrie de Fibonacci és 1, 2, 3, 5, 8, ... Cada nombre de la seqüència (excepte els dos primers) és el resultat de la suma dels dos anteriors. Fer un script en python que calculi el nombre n-èssim de Fibonacci, sent n un paràmetre entrat per l'usuari.

```
num=input("Quin nombre de fibbo?")
fib1=1
fib2=2
for i in range(3,num+1):
 fib3=fib2
 fib2=fib1+fib2
 fib1=fib3
print "Resultat :", fib2
```

# Exercicis

- Exercici 1: La sèrie de Fibonacci és 1, 2, 3, 5, 8, ... Cada nombre de la seqüència (excepte els dos primers) és el resultat de la suma dels dos anteriors. Fer un script en python que calculi el nombre n-èssim de Fibonacci, sent n un paràmetre entrat per l'usuari.

```
num=input("Quin nombre de fibbo?")
fib1=1
fib2=2
for i in range(3,num+1):
 fib1, fib2=fib2, fib1+fib2
print "Resultat :", fib2
```

# Exercicis

- Exercici 2: Calcular  $n!/(n-m)!m!$  A partir d'una funció que rebi m i n (combinacions(m,n)), tenint en compte que n ha de ser més gran o igual que m.

```
def factorial(n):
```

```
 If n==0:
```

```
 return 1
```

```
 else:
```

```
 return n*factorial(n-1)
```

```
def combinacions(m,n):
```

```
 if m<n:
```

```
 return factorial(n)/(factorial(n-m)*factorial(m))
```

# Exercicis

- Exercici 3: Dissenya una classe anomenada Set que representi el concepte de conjunt. Aquesta classe hauria de tenir els següents mètodes associats:
  - Set(elements): Crea un conjunt ('elements' és una llista que conté els elements).
  - addElement(x): Afegeix x al conjunt.
  - deleteElement(x): Elimina x del conjunt.
  - member(x): Retorna True si és membre del conjunt.
  - intersection(set2): Retorna un nou conjunt que conté els elements comuns amb set2.
  - union(set2): Retorna un nou conjunt que conté els elements que estan al conjunt, a set2, o a ambdós conjunts.
  - subtract(set2): Retorna un nou conjunt que conté tots els elements que no són a set2.



# Exercicis

- Exercici 3:

Class Set:

```
def __init__(Self,elem):
 Self.elements=elem
def addElement(Self,x):
 Self.elements.append(x)
def deleteElement(Self,x):
 Self.elements.remove(x)
def member(Self,x):
 x in Self.elements
def intersection(Self,set2):
 comuns=[]
 for valors in Self.elements:
 if valors in set2.elements:
 comuns.append(valors)
 comu=Set(comuns)
 return comu
```

```
def union(Self,set2):
 comuns=[]
 for valors in Self.elements:
 if not valors in comuns:
 comuns.append(valors)
 for valors in set2.elements:
 if not valors in comuns:
 comuns.append(valors)
 comu=Set(comuns)
 return comu
def substract(Self,set2):
 comuns=[]
 for valors in Self.elements:
 if not valors in set2.elements:
 comuns.append(valors)
 comu=Set(comuns)
 return comu
```

# Monitorització del sistema

Administració de Sistemes Operatius  
Dept. Matemàtica Aplicada i Anàlisi,  
Universitat de Barcelona

# Index

- Introducció a la Monitorització
- Control de la carrega de treball
  - Processos
  - CPU
  - Memòria
- Monitorització del sistema de fitxers
  - Entrada i sortida
  - Ocupació de disc
- Monitorització d'usuaris i logs
- Monitorització de la xarxa
- Tasques automàtiques
- Accounting

# Introducció

- Què és la monitorització?
  - Procés de supervisió necessari per a l'execució d'un pla encaminat a aconseguir una meta determinada.
  - Detectar possibles problemes a temps, i corregir-los adequadament.
- La pregunta clau: **Per què va tot tant lent?**

# Introducció

- A part de la **detecció de possibles problemes**, la monitorització té la missió d'identificar quins són els actuals colls d'ampolla = **On falten els recursos?**
  - Comprar-ne més per augmentar el rendiment.
  - Mesurar i eliminar les tasques que fan ús ineficient del sistema.
  - Ex: CPU insuficient:
    - Eliminar tasques innecessàries.
    - Programar tasques no prioritàries per a que s'executin en hores de poca càrrega.
    - Limitar el nombre de tasques que corren alhora.

# Introducció

- A continuació veurem els mètodes que ens proporcionen els sistemes operatius per a engegar tasques automàticament, executant-les en un determinat instant de temps, o periòdicament.
- Finalment, veurem el concepte d'accounting o mesura del volum de recursos consumits per part d'un determinat usuari.

# Control de la càrrega de treball

1. Identificar el problema i descriure'l formalment sense ambigüitats.
  - El programa XXX funciona lentament quan obro 3 finestres i obro un fitxer de més de YYY Mb.
2. Esbrinar que més s'està executant al sistema.
3. Formular detalladament els objectius de rendiment desitjats i quines aplicacions són prioritàries.
4. Dissenyar les modificacions que permeten assolir els objectius.
5. Monitoritzar el sistema per comprovar si els canvis funcionen.
6. Tornar al punt 1. (no s'acaba mai...)

**SI UNA COSA FUNCIONA, NO LA TOQUIS**

# Control de la càrrega de treball

## Monitorització de processos

- UNIX permet establir prioritats per afavorir certs processos.
- Eines i Utilitats:
  - **uptime**: permet estimar de forma aproximada la càrrega del sistema.

```
>uptime
```

```
0:23pm up 4 days, 32 users, load average 2.1, 1.67, 1.12
```

- Retorna l'hora, les hores que el sistema ha estat engegat, i el nombre promig de processos executant-se l'últim minut, i els últims 5 i 15 minuts.



# Control de la càrrega de treball

## Monitorització de processos

- Eines i Utilitats:

- **ps**: Mostra una estadística dels processos que corren actualment al sistema.

```
>ps aux
```

```
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
root 1 45.4 0.0 1584 264 ? S Jun12 0:18 init [2]
root 2 23.1 0.0 0 0 ? S Jun12 0:04 [migration/0]
root 3 10.9 0.0 0 0 ? SN Jun12 0:00 [ksoftirqd/0]
root 4 2.0 0.0 0 0 ? S Jun12 0:03 [migration/1]
```

- Retorna una llista dels processos ordenats de més a menys consum de CPU. VSZ (Memoria Virtual usada), RSS (memòria física usada), STAT (estat del procés), START TIME (hora en que es va iniciar el procés).

# Control de la càrrega de treball

## Monitorització de processos

- Eines i Utilitats:

- Es poden fer cerques combinant amb la utilitat grep:

- ```
>ps aux | grep usuari
```

- Altres utilitats per llistar l'arbre de processos:

- ```
>pstree -A
```

```
init-+-apache---5*[apache]
 |-apache-ssl-+-apache-ssl
 | `--gcache
 |-apcupsd
 |-atd
 |-automount
 |-cron-+-cron---mdebian---rsync-deb---rsync
 | `--cron---sh---analog.monthly---analog
 |-cupsd
```

# Control de la càrrega de treball

## Monitorització de processos

- Eines i Utilitats:
  - Utilitat **top**: Llista els processos que tenen més consum de CPU en temps real. La llista es va actualitzant contínuament.

```
Tasks: 46 total, 1 running, 45 sleeping, 0 stopped, 0 zombie
Cpu(s): 1.0% user, 2.3% system, 0.0% nice, 96.8% idle
Mem: 127448k total, 105844k used, 21604k free, 23884k buffers
Swap: 128480k total, 0k used, 128480k free, 52164k cached
```

| PID | USER   | PR | NI | UIRT | RES  | SHR  | S | %CPU | %MEM | TIME+   | COMMAND        |
|-----|--------|----|----|------|------|------|---|------|------|---------|----------------|
| 953 | root   | 16 | 0  | 1044 | 1044 | 848  | R | 3.5  | 0.8  | 0:00.31 | top            |
| 1   | root   | 9  | 0  | 500  | 500  | 448  | S | 0.0  | 0.4  | 0:00.25 | init           |
| 2   | root   | 8  | 0  | 0    | 0    | 0    | S | 0.0  | 0.0  | 0:00.00 | keventd        |
| 3   | root   | 19 | 19 | 0    | 0    | 0    | S | 0.0  | 0.0  | 0:00.00 | ksoftirqd_CPU0 |
| 4   | root   | 9  | 0  | 0    | 0    | 0    | S | 0.0  | 0.0  | 0:00.00 | kswapd         |
| 5   | root   | 9  | 0  | 0    | 0    | 0    | S | 0.0  | 0.0  | 0:00.00 | bdflush        |
| 6   | root   | 9  | 0  | 0    | 0    | 0    | S | 0.0  | 0.0  | 0:00.21 | kupdated       |
| 99  | root   | 9  | 0  | 0    | 0    | 0    | S | 0.0  | 0.0  | 0:00.60 | kjournald      |
| 460 | root   | 9  | 0  | 0    | 0    | 0    | S | 0.0  | 0.0  | 0:00.00 | khubd          |
| 547 | daemon | 9  | 0  | 452  | 452  | 384  | S | 0.0  | 0.4  | 0:00.02 | portmap        |
| 711 | root   | 9  | 0  | 808  | 808  | 704  | S | 0.0  | 0.6  | 0:00.33 | syslogd        |
| 714 | root   | 9  | 0  | 1388 | 1384 | 480  | S | 0.0  | 1.1  | 0:01.00 | klogd          |
| 722 | bind   | 9  | 0  | 2596 | 2592 | 1944 | S | 0.0  | 2.0  | 0:00.03 | named          |
| 725 | bind   | 9  | 0  | 2596 | 2592 | 1944 | S | 0.0  | 2.0  | 0:00.00 | named          |
| 726 | bind   | 9  | 0  | 2596 | 2592 | 1944 | S | 0.0  | 2.0  | 0:00.14 | named          |
| 727 | bind   | 9  | 0  | 2596 | 2592 | 1944 | S | 0.0  | 2.0  | 0:00.00 | named          |
| 728 | bind   | 9  | 0  | 2596 | 2592 | 1944 | S | 0.0  | 2.0  | 0:00.00 | named          |
| 734 | root   | 9  | 0  | 2488 | 2484 | 1780 | S | 0.0  | 1.9  | 0:00.02 | lwresd         |

```
debian:~# _
```

# Control de la càrrega de treball

## Monitorització de processos

- Eines i Utilitats: **El /proc filesystem**

- És un sistema de fitxers virtual, on els fitxers que s'hi troben són parts de la memòria del kernel i les seves estructures.
- Estructurat en forma de directoris que porten el nom del PID de cada procés:

```
>cat /proc/cpuinfo
```

```
processor fd0v_bug : no
vendor_id h1GebugneIntel : no
cpu family f00f_bug : no
model c07a_bug : no
model name fpPentium III (Katmai)
stepping fp3_exception : yes
cpu MHz cp500.200el : 2
cache size wp512 KB : yes
```

# Control de la càrrega de treball

## Monitorització de processos

- Eines i Utilitats: **El /proc filesystem**
  - Fitxers globals
    - /proc/cpuinfo: Informació del processador.
    - /proc/devices: Informació dels dispositius (major/minor number).
    - /proc/filesystems: Sistemes de fitxers suportats.
    - /proc/meminfo: Percentatge d'ús de memòria.
    - /proc/modules: Mòduls carregats al kernel.
    - /proc/pci: Targes PCI detectades i la seva configuració.
    - /proc/scsi: Dispositius SCSI detectats.
    - /proc/version: la versió actual de linux.

# Control de la càrrega de treball

## Monitorització de processos

- Límits de recursos pels processos
  - Dos tipus de límits:
    - Límits actuals per defecte. (`ulimit -a`)
    - Límits absoluts (hard límits) (`ulimit -Ha`)
  - Els podem conèixer i modificar mitjançant l'eina `ulimit`:

```
>ulimit -a
core file size (blocks, -c) 0
data seg size (kbytes, -d) unlimited
file size (blocks, -f) unlimited
max locked memory (kbytes, -l) unlimited
max memory size (kbytes, -m) unlimited
open files (-n) 1024
pipe size (512 bytes, -p) 8
stack size (kbytes, -s) 8192
cpu time (seconds, -t) unlimited
max user processes (-u) unlimited
virtual memory (kbytes, -v) unlimited
```

# Control de la càrrega de treball

## Monitorització de processos

- Matar processos

- Comanda **kill**: Envia una senyal a un procés/processos

```
kill [-senyal] pids
```

- Exemple: `kill -15 234` o bé `kill -9 234`

- Matar múltiples processos amb **killall**:

```
killall -KILL comanda
```

- Altres senyals:

- QUIT (3) o INT (12) per matar processos que no volen morir (estan fent I/O, NFS,...).
- STOP i CONT: Aturar i engegar un procés.

# Control de la càrrega de treball

## Monitorització de la CPU

- Sistemes UNIX: Política Round Robin amb una prioritat associada a cada procés.
  - Quan s'allibera la CPU, entra el procés que té menor valor de prioritat.
  - Un procés s'executa fins que: acaba, fa I/O, rep un senyal o acaba el quantum de CPU.
  - Dos tipus de número que indiquen prioritat:
    - Nice number: Prioritat d'execució respecte altres processos.
    - Prioritat Actual: Calculada dinàmicament, tenint en compte el nice number, el temps de CPU que ha tingut recentment, quin altres processos estan corrent, etc.



# Control de la càrrega de treball

## Monitorització de la CPU

- Nice number especial: 20. Només per aquells processos que s'han d'executar quan no n'hi ha cap altre.
  - Tots els usuaris poden baixar la prioritats dels seus processos (augmentant el nice number).
  - Només root pot pujar les prioritats dels processos (disminuint el nice number).
  - Modificar les prioritats:

```
>nice [+|-] process
>nice +6 calcula_trajectories
>ps 1 #mirar el NI modificat
```

# Control de la càrrega de treball

## Monitorització de la CPU

- Conèixer l'estadística d'ús de la CPU:
  - Utilitat **vmstat**:  
>vmstat interval [nvegades]
  - Mostra el percentatge d'ús de cpu durant *interval* segons, cada *nvegades*.

```
>vmstat 5 4
```

```
procs -----memory----- -swap- ---io--- --system-- ----cpu----
 r b swpd free buff cache si so bi bo in cs us sy id wa
 0 0 78420 8420 107284 199640 0 0 3 0 1 1 1 1 95 3
 0 2 78420 8436 107296 199628 0 0 0 15 261 40 0 0 99 0
 0 0 78420 8444 107304 199620 0 0 0 18 270 32 0 0 99 1
 0 0 78420 8444 107320 199604 0 0 0 12 254 31 0 0 99 1
```

# Control de la càrrega de treball

## Monitorització de la CPU

```
>vmstat 5 4
```

```
procs -----memory----- -swap- ---io--- --system-- ----cpu----
r b swpd free buff cache si so bi bo in cs us sy id wa
0 0 78420 8420 107284 199640 0 0 3 0 1 1 1 1 95 3
0 2 78420 8436 107296 199628 0 0 0 15 261 40 0 0 99 0
0 0 78420 8444 107304 199620 0 0 0 18 270 32 0 0 99 1
0 0 78420 8444 107320 199604 0 0 0 12 254 31 0 0 99 1
```

r: Número de processos esperant CPU.

cs: Número de canvis de context.

us: Percentatge de cicles de CPU dedicats a aplicacions d'usuari.

sy: Percentatge d'ús de la CPU consumit pel sistema.

Id: Percentatge de CPU que ha quedat lliure (*idle*).

# Control de la càrrega de treball

## Monitorització de la CPU

- Indicadors de mancances de CPU:
  - Càrrega de treball superior a la mitja habitual.
  - Quan l'ús (% cicles usuari) sigui més alt del normal.
  - Hi hagi molts processos esperant CPU .
  - Quan hi hagi molt percentatge de CPU de sistema o molts canvis de context.
- Accions:
  - Reestructurar les prioritats dels processos.
  - Moure processos a altres sistemes o executar-los en hores de baix consum.
  - Canviar el planificador per a que atengui als processos que creiem adequats.

# Control de la càrrega de treball

## Monitorització de la CPU

- Reestructurar les prioritats dels processos.
  - Baixar la prioritat a processos que consumeixen molt.
  - Pujar la prioritat a processos crítics.
  - Comanda: **renice** *nice\_number pid\_proceso*  
>renice 10 20438

# Control de la càrrega de treball

## Monitorització de la memòria

- Els efectes de l'ús de la memòria tenen una incidència igual o major que la CPU.
- Necessitats de memòria:
  - Pocs processos molt grans.
  - Sistemes amb molts processos petits.
- Quan les necessitats excedeixen els límits de memòria física: *paginació* i *swapping*
  - Paginació: Moure fragments o “pàgines” d'un procés a disc per alliberar memòria.
  - Swapping: Escriure un procés sencer a disc, alliberant tota la memòria física que ocupa.

# Control de la càrrega de treball

## Monitorització de la memòria

- Avantatges de la paginació:
    - Permet tenir un espai de memòria virtual molt més gran que la memòria física disponible.
    - Aprofita el principi de la localitat dels programes.
  - Problemes de la paginació:
    - Quan hi ha poca memòria física per a les necessitats del sistema, és possible que s'estigui contínuament enviant i portant pàgines a disc.
- ↓
- **Consum de recursos en paginació en comptes de càlcul.**

# Control de la càrrega de treball

## Monitorització de la memòria

- Utilitats: free
  - Determinar la quantitat de memòria del sistema:

```
>free
 total used free shared buffers cached
Mem: 516080 499388 16692 0 57984 280188
-/+ buffers/cache: 161216 354864
Swap: 530104 78420 451684
```



# Control de la càrrega de treball

## Monitorització de la memòria

- Utilitats: top
  - Determinar la quantitat de memòria disponible/usada al sistema

```
Tasks: 46 total, 1 running, 45 sleeping, 0 stopped, 0 zombie
Cpu(s): 1.0% user, 2.3% system, 0.0% nice, 96.8% idle
```

| PID | USER   | PR | NI | UIRT | RES  | SHR  | S | %CPU | %MEM | TIME+   | COMMAND        |
|-----|--------|----|----|------|------|------|---|------|------|---------|----------------|
| 953 | root   | 16 | 0  | 1044 | 1044 | 848  | R | 3.5  | 0.8  | 0:00.31 | top            |
| 1   | root   | 9  | 0  | 500  | 500  | 448  | S | 0.0  | 0.4  | 0:00.25 | init           |
| 2   | root   | 8  | 0  | 0    | 0    | 0    | S | 0.0  | 0.0  | 0:00.00 | keventd        |
| 3   | root   | 19 | 19 | 0    | 0    | 0    | S | 0.0  | 0.0  | 0:00.00 | ksoftirqd_CPU0 |
| 4   | root   | 9  | 0  | 0    | 0    | 0    | S | 0.0  | 0.0  | 0:00.00 | kswapd         |
| 5   | root   | 9  | 0  | 0    | 0    | 0    | S | 0.0  | 0.0  | 0:00.00 | bdflush        |
| 6   | root   | 9  | 0  | 0    | 0    | 0    | S | 0.0  | 0.0  | 0:00.21 | kupdated       |
| 99  | root   | 9  | 0  | 0    | 0    | 0    | S | 0.0  | 0.0  | 0:00.60 | kjournald      |
| 460 | root   | 9  | 0  | 0    | 0    | 0    | S | 0.0  | 0.0  | 0:00.00 | khubd          |
| 547 | daemon | 9  | 0  | 452  | 452  | 384  | S | 0.0  | 0.4  | 0:00.02 | portmap        |
| 711 | root   | 9  | 0  | 808  | 808  | 704  | S | 0.0  | 0.6  | 0:00.33 | syslogd        |
| 714 | root   | 9  | 0  | 1388 | 1384 | 480  | S | 0.0  | 1.1  | 0:01.00 | klogd          |
| 722 | bind   | 9  | 0  | 2596 | 2592 | 1944 | S | 0.0  | 2.0  | 0:00.03 | named          |
| 725 | bind   | 9  | 0  | 2596 | 2592 | 1944 | S | 0.0  | 2.0  | 0:00.00 | named          |
| 726 | bind   | 9  | 0  | 2596 | 2592 | 1944 | S | 0.0  | 2.0  | 0:00.14 | named          |
| 727 | bind   | 9  | 0  | 2596 | 2592 | 1944 | S | 0.0  | 2.0  | 0:00.00 | named          |
| 728 | bind   | 9  | 0  | 2596 | 2592 | 1944 | S | 0.0  | 2.0  | 0:00.00 | named          |
| 734 | root   | 9  | 0  | 2488 | 2484 | 1780 | S | 0.0  | 1.9  | 0:00.02 | lwresd         |

```
debian:~# _
```

# Control de la càrrega de treball

## Monitorització de la memòria

- Utilitats: vmstat
  - Informació més detallada del sistema de memòria

```
>vmstat 5 4
```

```
procs -----memory----- -swap- ---io--- --system-- ----cpu----
r b swpd free buff cache si so bi bo in cs us sy id wa
0 0 78420 8420 107284 199640 0 0 3 0 1 1 1 1 95 3
0 2 78420 8436 107296 199628 0 0 0 15 261 40 0 0 99 0
0 0 78420 8444 107304 199620 0 0 0 18 270 32 0 0 99 1
0 0 78420 8444 107320 199604 0 0 0 12 254 31 0 0 99 1
```

- Grups de columnes 2,3,4 i 5.
  - Memory: Ús de dades
  - Page swap: Activitat de paginació (si=desde disc) (so=al disc)
  - Io: Operacions I/O. (bi/bo=blocs rebuts/enviats a un dispositiu)
  - System: Nombre de interrupcions i canvis de context per segon.

# Control de la càrrega de treball

## Monitorització de la memòria

- Interpretar quan hi ha problemes de memòria:
  - La memòria disponible cau per sota un llindar (5-15%).
  - Activitat important de paginació a disc. (*columna so, si no és significatiu: pot provenir de carregar nous processos*).
  - El sistema hiperpagina sovint (ni que sigui per poc temps).
  - El fitxer de paginació incrementa en excés la seva mida.

# Control de la càrrega de treball

## Monitorització de la memòria

- Configuració del gestor de memòria
  - Els sistemes UNIX permeten variar la configuració del gestor de memòria.
  - LINUX: Es fa usant el `/proc/sys/vm`
  - Fitxers especials (modificar només per experts):
    - Freepages: Mínim número de pàgines lliures que ha de tenir el sistema.
    - Buffermem: Memòria destinada a la cache del sistema de fitxers.
    - overcommit\_memory: Permet a aplicacions reservar més memòria de la que hi ha (mai l'arriben a usar).

# Control de la càrrega de treball

## Monitorització de la memòria

- Gestionar l'espai de paginació:
  - Als sistemes UNIX hi acostuma a haver-hi una partició o un fitxer dedicat a *swapping*.
  - Quantitat d'espai dedicat a *swapping*? Depén...
    - Workstations o sistemes petits: 1-2 cops l'àrea de memòria física.
    - Sistemes grans 2-3 vegades.
  - Alguns operatius no reserven de forma efectiva grans quantitats d'espai fins que no és necessari.

# Control de la càrrega de treball

## Monitorització de la memòria

- Gestionar l'espai de paginació:

- Utilitats

- Informació sobre l'àrea de paginació:

```
>cat /proc/swaps
```

| Filename  | Type      | Size   | Used  | Priority |
|-----------|-----------|--------|-------|----------|
| /dev/hda1 | partition | 530104 | 78420 | -1       |

- Mostrar l'actual estadística d'ús:

```
>swapon -s o bé
```

```
>free -m -o
```

|       | total | used | free | shared | buffers | cached |
|-------|-------|------|------|--------|---------|--------|
| Mem:  | 503   | 497  | 6    | 0      | 64      | 289    |
| Swap: | 517   | 76   | 441  |        |         |        |

# Control de la càrrega de treball

## Monitorització de la memòria

- Gestionar l'espai de paginació:
  - Activació de l'àrea de swap
    - Normalment al boot, definint l'àrea de swap a `/etc/fstab`

```
/dev/hda1 none swap sw 0 0
```
    - Comanda en temps de boot per activar la swap:

```
>swapon -a >/dev/console 2>&1
```

# Control de la càrrega de treball

## Monitorització de la memòria

- Gestionar l'espai de paginació:
  - Creació d'una nova àrea de swap
    - Opció 1: Fer un back up del disc, eliminar les particions, i fer un nou sistema de fitxers on hi càpiga la nova àrea de swap,... **costós!**
    - Establir l'àrea de swap en el mateix sistema de fitxers mitjançant un arxiu (és una mica més lent):

`Utilitat mkswap`



# Control de la càrrega de treball

## Monitorització de la memòria

- Gestionar l'espai de paginació:
  - Priorització de l'àrea de swap
    - swap -p, otorga una prioritat a cada partició.
      - S'acostuma a donar menys prioritat a les particions de swap muntades en sistema de fitxers.
  - Eliminar una àrea de swap:
    - Eliminar la línia de /etc/fstab.
    - Reboot del sistema (cal garantir que no hi ha cap fragment en ús).
    - rm del fitxer associat.

# Monitorització del sistema de fitxers

- Les eines per a la monitorització dels sistemes d'accés a disc en UNIX són una mica pobres.

Eina sar: `sar -d interval [nvegades]`

```
>sar
```

| 17:58:15 | %usr | %sys | %wio | %idle |
|----------|------|------|------|-------|
| 17:58:17 | 43   | 9    | 1    | 46    |
| 17:58:19 | 35   | 17   | 3    | 45    |
| 17:58:21 | 36   | 22   | 20   | 23    |
| 17:58:23 | 21   | 17   | 0    | 63    |
| 17:58:25 | 85   | 12   | 3    | 0     |
| Average  | 44   | 15   | 5    | 35    |

# Monitorització del sistema de fitxers

- Optimització del sistema de fitxers:
  - Acostuma a ser més factible en temps de creació del sistema de fitxers.
  - Tipus de sistemes I/O:
    - Accés seqüencial: Accés als blocs en ordre, un darrere l'altre. Només limitat per la capacitat de transferència del disc.
    - Accés aleatori: Les dades es poden llegir en qualsevol ordre. Cal minimitzar el temps de posicionament.

# Monitorització del sistema de fitxers

- Optimització del sistema de fitxers:
  - El rendiment del disc esta condicionat per:
    - El propi Hardware del disc.
    - La distribució de les dades en el sistema de discs.
    - La localització física de les dades al disc.

# Monitorització del sistema de fitxers

- Optimització del sistema de fitxers: **Hardware.**
  - SCSI >> EIDE. Però més cars
  - Altres factors: Cache local al disc.
  - Velocitat de transferència màxima.
  - Accés aleatori: temps promig d'accés + rpm.
  - Qualitat de la controladora.
  - No saturar la controladora.
    - Connectar un disc a cada controladora si és possible.

# Monitorització del sistema de fitxers

- Optimització del sistema de fitxers: **Distribució de les dades en el sistema de discs.**
  - Distribuir els arxius en temps de planificació de les particions de manera que es reparteix al màxim els accessos de les controladores. Ex:
    - Càrrega principalment d'usuari: Distribuir el /home en múltiples discs.
    - Sistema de BD, distribuir les taules on es fa un màxim accés en diferents discs.

# Monitorització del sistema de fitxers

- Optimització del sistema de fitxers: **Localització física de les dades.**
  - L'accés seqüencial és MOLT més eficient si els fitxers estan situats de forma contígua.
  - L'accés és més ràpid a la part més externa del disc.
  - Disc "stripping" es beneficiós en fitxers grans.
  - Accés aleatori es més ràpid quan les dades són al centre del disc.
  - Defragmentació de disc.

# Monitorització del sistema de fitxers

- Ajustar el rendiment del sistema I/O:
  - L'accés read-ahead (“llegir per endavant”): Quan el SO detecta que s'està fent una lectura seqüencial, pot llegir les pàgines següents, sense que el procés ho hagi demanat. Heurísticament es minimitza el temps per a futurs accessos.

S'implementa mitjançant el `/proc/sys/vm`

- `Page-cluster`: Número de pàgines llegides per una sola operació de lectura. Es va incrementant a mesura que es fan més accessos seqüencials.
- `Min-readahead` i `max-readahead`: Estableix el màxim i mínim número de pàgines que es poden portar per readahead.



# Monitorització del sistema de fitxers

- Ajustar el rendiment del sistema I/O:
    - L'accés retardat a disc: Consisteix en posar una cua per a que els accessos llargs a disc no alenteixin la interactivitat del sistema.
      - Deshabilitat per defecte.
    - Activable mitjançant la utilitat: `chdev`  
`>chdev -l sys0 -a maxpout=33 -a minpout=16`
- Si s'intenta escriure en un disc on hi ha més de 33 peticions, s'haurà d'esperar fins que n'hi hagi menys de 16.

# Monitorització del sistema de fitxers

- Monitorització i gestió de l'ús de disc:
  - Utilitat df: Dona una mesura de l'espai emprat i disponible a cada disc:

```
>df -k
Filesystem 1K-blocks Used Available Use% Mounted on
/dev/hda5 303344 87086 200597 31% /
tmpfs 258040 0 258040 0% /dev/shm
/dev/hda2 69995 8365 58016 13% /boot
/dev/hda6 1161068 1082460 19572 99% /var
/dev/hda3 2071416 1492324 473868 76% /usr
/dev/sdb4 4988880 32860 4702596 1% /images
/dev/hdd1 192292124 50257556 132266648 28% /home
• /dev/hdc1 192292124 131426888 51097316 73% /homes2
```

Hi ha un 5-10% d'espai que el sistema reserva i no apareix a les estadístiques.

# Monitorització del sistema de fitxers

- Monitorització i gestió de l'ús de disc:
  - Utilitat du: Dona una mesura de l'espai emprat en un directori i subdirectoris:

```
>du -k /home/sergio
4 /home/sergio/Mail
38284 /home/sergio/mail
8 /home/sergio/.ssh
272 /home/sergio/Amics
5792 /home/sergio/Papers
64 /home/sergio/public_html/images
160 /home/sergio/public_html
2360760 /home/sergio
```

# Monitorització del sistema de fitxers

- Monitorització i gestió de l'ús de disc:
  - Utilitat quot: Retorna l'espai usat per cada usuari en el sistema de fitxers especificat:

```
>quot /
4356 root
19832 david
124 joan
128 sys
4 adm
```

# Monitorització del sistema de fitxers

- Monitorització i gestió de l'ús de disc:
  - Solucionar els problemes d'espai:
    - Comprar un altre disc.
    - Muntar un disc remot.
    - Eliminar fitxers innecessaris.
    - Comprimir els fitxers grans, accedits poc freqüentment.
    - Convèncer als usuaris que facin còpia de seguretat i esborrin els fitxers innecessaris.
    - Polítiques dràstiques: fer un backup i eliminar els fitxers d'usuari que no s'hagin tocat ens els últims mesos.
    - Eliminar fitxers de log que han crescut en excés.



# Monitorització del sistema de fitxers

- Monitorització i gestió de l'ús de disc:
  - Solucionar els problemes d'espai:
    - Sovint cal buidar el directori /tmp.
    - El /var/spool o /usr/spool pot tenir treballs d'impressió no actius que ocupen espai.
    - Utilitat find: busca fitxers segons un criteri:  
>find / -name ".BAK.\*"  
Busca fitxers que poden ser còpies de seguretat o arxius temporals de editors de text, etc...

# Monitorització del sistema de fitxers

- Monitorització i gestió de l'ús de disc:

- Limitar la mida dels fitxers de log.

- Solen estar a `/var/adm` o `/usr/adm`

- Alguns defineixen el seu path a `/etc/syslog.conf`

- Buidar un fitxer de log:

- ```
>cat /dev/null > /var/adm/sulog
```

Es preferible a eliminar-lo, ja que algunes utilitats no el tornen a crear.

- Mantenir només la darrera part del fitxer de log:

- ```
>tail -100 sulog >tmp
```

- ```
>cat tmp>sulog
```

Monitorització del sistema de fitxers

- Monitorització i gestió de l'ús de disc:
 - Limitar l'ús de disc dels usuaris: quotes de disc.
 - Hard limit: L'usuari mai el pot passar.
 - Soft limit: L'usuari el pot passar i se'l advertirà fins que redueixi el seu espai.
 - Monitorització de les quotes de disc: utilitat repquota.

```
>repquota -v /system
```

```
          Bloc limits      file limits
User      used sof  hard  grace used soft hard grace
Usuari 14578 30000 35000    3    843  0    0
```

Més informació al "Gestió de recursos"

Monitorització d'usuari i logs

- ¿qui fa ús del sistema?
 - who, w, finger
 - last --> últimes entrades al sistema
 - lastlog --> últim login per cada usuari (logins erronis: /var/log/auth.log)

Monitorització d'usuaris i logs

- ¿On indiquem qué registrar?
- Missatges del sistema
 - Quan hi ha problemes en el funcionament d'algun servei, el sistema guarda uns log.
 - Comprovació automàtica dels fitxers de log.
 - Eliminació de la informació de log caducada.
- `/etc/syslog.conf`

Monitorització d'usuaris i logs

- `/etc/syslog.conf`
- Format: `facilitat.nivell fitxer_desti`
 - Facilitat és el servei: kern (kernel), mail, lpr (impressió), ftp, etc...
 - Nivell: Nivell de seguretat. Ex: emerg, alert, crit, err, info, debug, none, mark (timestamps)

Exemple: `/etc/syslog.conf`.

```
cron.*                /var/log/cron.log
daemon.*              /var/log/daemon.log
kern.*                /var/log/kern.log
lpr.*                 /var/log/lpr.log
```

- Programes **syslogd** **logd**:

Redirigeixen seguint les indicacions a `/etc/syslog.conf`.

–

Monitorització d'usuari i logs

- Fer logs per monitoritzar
- Alguns:
 - /var/log/messages --> missatges informatius diversos
 - /var/log/auth.log --> missatges autenticació
 - /var/log/syslog
 - /var/log/cron.log
 - /var/log/daemon.log --> missatges variis
 - /var/log/kern.log --> missatges del kernel
 - /var/log/lpr.log --> missatges sobre impressores
 - /var/log/mail.log --> missatges correu
 - /var/log/user.log --> missatges diversos tipus user
 - /var/log/uucp.log

Monitorització de la xarxa

- La monitorització del rendiment del sistema de xarxa pot arribar a ser molt complexa.
- Utilitat bàsica: netstat, informació desde l'últim boot.

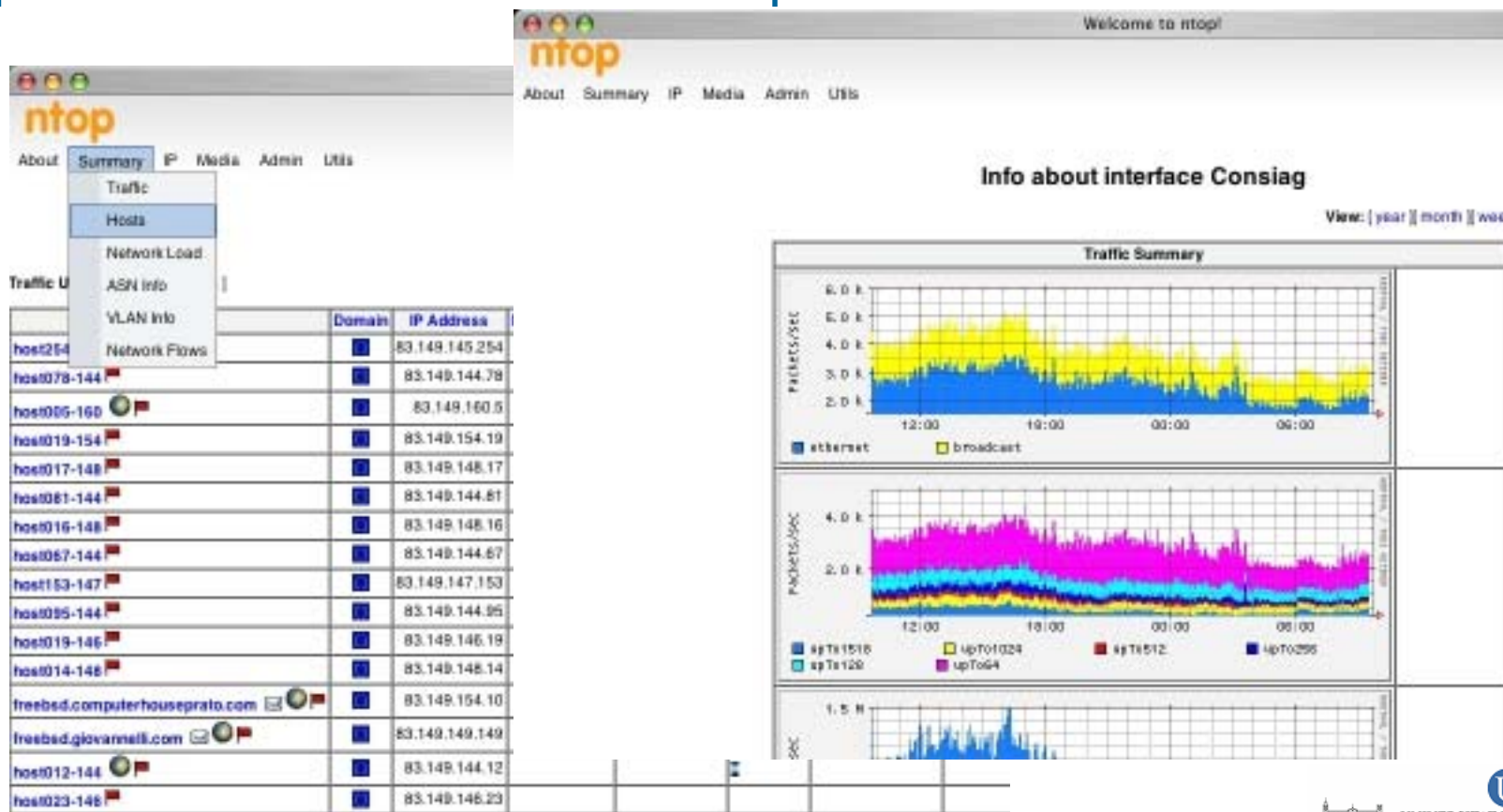
```
>netstat -s -p tcp (sortida restringida a tcp)
```

```
Tcp:
```

```
184684 active connections openings
438509 passive connection openings
927 failed connection attempts
77999 connection resets received
5 connections established
139607890 segments received
183316525 segments send out
340432 segments retransmitted
75 bad segments received.
123055 resets sent
```

Monitorització de la xarxa

- Paràmetres del rendiment de les xarxes TCP/IP
- Utilitats gràfiques per a la monitorització dels paràmetres de la xarxa: ntop.



Monitorització de la xarxa

- Paràmetres del rendiment de les xarxes TCP/IP
- Monitorització bàsica de connectivitat i temps de resposta: Utilitat ping.

```
>ping -c 5 www.google.com
```

```
PING www.l.google.com (66.249.85.99): 56 data bytes
```

```
64 bytes from 66.249.85.99: icmp_seq=0 ttl=242 time=52.6 ms
```

```
64 bytes from 66.249.85.99: icmp_seq=1 ttl=242 time=53.9 ms
```

```
64 bytes from 66.249.85.99: icmp_seq=2 ttl=242 time=53.4 ms
```

```
64 bytes from 66.249.85.99: icmp_seq=3 ttl=242 time=52.5 ms
```

```
64 bytes from 66.249.85.99: icmp_seq=4 ttl=242 time=53.7 ms
```

```
--- www.l.google.com ping statistics ---
```

```
5 packets transmitted, 5 packets received, 0% packet loss
```

```
round-trip min/avg/max = 52.5/53.2/53.9 ms
```

Monitorització de la xarxa

- Paràmetres del rendiment de les xarxes TCP/IP
- Determinar la ruta que segueixen els paquets: Utilitat traceroute.

```
>traceroute www.google.com
```

```
traceroute: Warning: www.google.com has multiple addresses; using  
66.249.85.104
```

```
traceroute to www.l.google.com (66.249.85.104), 30 hops max, 38  
byte packets
```

```
1  e3.facultat.mat.com.ub.es (161.116.4.254) 0.613ms 0.738ms  
0.540ms
```

```
2  10.14.201.254 (10.14.201.254) 0.434ms 0.411ms 0.495ms
```

```
3  10.1.5.10 (10.1.5.10) 0.661ms 0.475ms 0.457ms
```

```
...
```

```
29  * * * #Gateways sota firewall no visibles
```

```
30  * * *
```


Monitorització de la xarxa

- Paràmetres del rendiment de les xarxes TCP/IP
- Analitzar el tràfic de la xarxa:
 - Utilitat tcpdump.
 - Permet veure el tràfic de la xarxa. Exemple: login

```
>tcpdump -e -t host host1
```

```
veron.1023> odin.login: S 768512:768512(0) win 4096 <mss 1024>
```

```
odin.login> veron.1023: S 947648:947648(0) ack 768513 win 4096 <mss 1024>
```

```
veron.1023 > odin.login: . ack 1 win 4096
```

```
veron.1023 > odin.login: P 1:2(1) ack 1 win 4096
```

```
odin.login > veron.1023: . ack 2 win 4096
```

```
veron.1023 > odin.login: P 2:21(19) ack 1 win 4096
```

```
odin.login > veron.1023: P 1:2(1) ack 21 win 4077
```

```
odin.login > veron.1023: P 2:3(1) ack 21 win 4077 urg 1
```

```
odin.login > veron.1023: P 3:4(1) ack 21 win 4077 urg 1
```

Monitorització de la xarxa

- **Paràmetres del rendiment de les xarxes TCP/IP**
 - Problemes en la interfície de xarxa. Deguts a hardware mal configurat.
 - Servidor sobrecarregat:
 - Carència de memòria.
 - Excés de tràfic.
 - Ample de banda de la xarxa insuficient: excès en time outs de les comunicacions:
 - Hardware nou.
 - Subdividir la xarxa en subxarxes.

Monitorització de la xarxa

- **Paràmetres del rendiment de les xarxes TCP/IP**
 - Gran nombre de paràmetres que hi intervenen
 - La Mida màxima de segment: Determina la mida del paquet més gran que es pot transmetre per la xarxa.
 - Segments més grans comporten menys transmissions.
 - Típicament 1460 bytes.
 - La mida dels buffers dels sockets. Un buffer ple provoca que les aplicacions no puguin enviar dades. Típicament 48-64 Kb.
 - 500 connexions x 64 x 2 (llegir i escriure) \approx 64Mb
 - Es modifiquen mitjançant els fitxers:

```
>echo valor > /proc/sys/net/core/rmem_max
>echo valor /proc/sys/net/core/wmem_max
>echo valor /proc/sys/net/ipv4/tcp_rmem
>echo valor /proc/sys/net/ipv4/tcp_wmem
```

Monitorització de la xarxa

- Rendiment del DNS
- Dos aspectes fonamentals:
 - Que el servidor tingui la suficient capacitat.
 - Més d'un servidor: la càrrega ha d'estar balancejada
 - Posar un servidor de dns diferent a cada grup d'usuaris al seu `/etc/resolv.conf`

Monitorització de la xarxa

- Rendiment del NFS
- Monitoritzar el rendiment i el tràfic del NFS.
 - Comanda: `nfsstat -rc` (`nfsstat -s` : Mostra la informació desde el punt de vista del servidor)

```
nfsstat -rc
Client rpc:
tcp: calls badxids badverfs timeouts newcreds
      9          0          0          0          0
udp: calls badxids badverfs timeouts newcreds retrans
     125955      3          0         342          0          0
      badcalls   timers      waits
           343          229          0
```

Timeouts: Operacions que han fallat perquè el server no responia a temps.

- Badfix: Respostes rebudes duplicades degut a operacions retransmeses.
 - Si Timeout >> Badfix: problema de la xarxa.
 - Sinó el problema és al servidor.

Monitorització de la xarxa

- **Rendiment del NFS**
- Millorar el rendiment de NFS:
 - Alliberar daemons NFS no utilitzats.
 - Usar hardware adequat als servidors NFS molt sol·licitats.
 - Tenir en compte el sistema de fitxers subjacent: fragmentació, mides, etc...
 - Evitar directoris remots amb un gran nombre de fitxers.
 - Evitar efectuar tasques innecessàries al sistema remot: compilacions, temporals, etc.

Monitorització de la xarxa

- Referències bibliogràfiques web addicionals de la monitorització en xarxa:
 - <http://docs.sun.com/app/docs/doc/816-4555/6maoquiq1?a=view>
 - Revistes: <http://www.sysadminmag.com/>
 - Tpcdump:
<http://www.iepm.slac.stanford.edu/monitoring/passive/tcpdump.html>
 - NFS: <http://h30097.www3.hp.com/tipnfs.html>

Tasques automàtiques

- Executar una sèrie de tasques de forma automàtica.
 - Eina CRON.
 - Es basa en el daemon crond.
 - Fitxer de configuració /etc/crontab
 - Conté una taula amb els processos que s'han de llençar i quan s'han de llençar.
 - Format:

minuts hores dia-del-mes mes dia-setmana comanda

Indica el moment en que s'executara una comanda:

0,15,30,45 * * * * echo "Hola" # s'executa cada quart d'hora

30 3-15 * * * echo "Hola" # s'executa a i mitja cada hora de 3 de la matinada a 3 de la tarde

Tasques automàtiques

- Millores al cron:
 - Afegir “passos” usant la /. Exemple:
5-10/2 #de 5 a 10 cada **dues** hores
 - Definir variables d'entorn dins del crontab: PATH, HOME, SHELL,...
- Normalment la taula del cron pròpia de cada usuari es guarda a: /var/spool/cron/crontabs/usuari.
- Usuaris típics:
 - root: Fer tasques de monitorització i administració
 - lp: Fer neteja dels fitxers que poden quedar després d'impressions fallides...

Tasques automàtiques

- Tasques administratives al cron:
- Normalment es creen uns scripts: daily, weekly, monthly on s'executen les tasques administratives del sistema.
- /etc/cron.daily
 - Eliminar el contingut de /tmp.
 - Llençar informació d'accounting.
 - Eliminar informació de log innecessària.
 - Capturar dades de utilització del disc i seguretat.
- /etc/cron.weekly:
 - Eliminar fitxers innecessaris molt antics.
 - Executar fsck -n per comprovar possibles problemes al disc.
 - Monitoritzar les característiques de seguretat dels comptes d'usuari.

Tasques automàtiques

- /etc/cron.monthly
 - Llistar els fitxers on no s'ha accedit en tot el més.
 - Generar els resums mensuals d'informació d'accounting.
 - Regenerar la base de dades del man.
- Cron i seguretat:
 - Garantir que els scripts executats pel cron són segurs.
 - Garantir que cap usuari no autoritzat executa scripts al cron.
 - cron.allow: Fitxer on hi ha els usuaris que poden executar cron.
 - cron.deny: Si no existeix el allow, tots els usuaris que no surten al cron.deny poden executar el cron.

Accounting

- El sistema operatiu guarda estadístiques d'ús sobre els recursos utilitzats per cada procés (memòria, CPU, I/O).
 - Objectius:
 - Cobrar a l'usuari pel servei.
 - Aprofitar la informació per a monitorització i auditoria de seguretat.
- Quan un procés acaba, el kernel escriu un fitxer binari a /var/adm amb la informació d'accounting.

Accounting

- Informació guardada per cada procés
 - Nom de la imatge. Ex: ls
 - Temps de CPU usat.
 - Temps trigat fins a finalitzar.
 - Hora d'inici.
 - Usuaris i grups associats.
 - Ús de memòria.
 - Número de caràcters llegits i escrits.
 - Número de blocs I/O llegits i escrits.
 - Consola on s'ha iniciat.
 - Flags.
 - Condició de sortida del procés.

Accounting

- El sistema d'accounting BSD.
 - Tasques:
 - Habilitar el sistema d'accounting i posar-lo als scripts del boot.
 - Extreure periòdicament la informació d'accounting.
 - Llençar els generadors d'informes.
 - Fitxers d'accounting
 - savacct: Fitxer standard d'accounting.
 - usracct: Fitxer d'usuari.
 - Utilitat: sa

Accounting

- Habilitar el sistema d'accounting i posar-lo als scripts del boot.
 - Utilitat accton: `accton [fitxer]`
 - Sinó se li subministra un fitxer on escriure les dades, no el crea.
 - Si es crida sense arguments inhabilita l'accounting.
- Extreure periòdicament la informació d'accounting.
 - Utilitat sa.

```
>cd /var/adm
```

```
>/usr/sbin/accton      #inhabilitar accounting temporalment
```

```
>mv fitxer_acc fixer_acc.sav
```

```
>cat /dev/null > fitxer_acc
```

```
>/usr/sbin/accton fitxer_acc #habilitem de nou
```

```
>sa -s fitxer_acc.sav    #Ho guarda a savacct i usracct
```

Accounting

- Llençar els generadors d'informes.
 - Utilitat sa: Llegeix les dades binaries i les mostra en informes.
 - Opcions:
 - -b: Promig de CPU per execució
 - -d Promig d'operacions de I/O
 - -D Total d'operacions de I/O
 - -k Ús de memoria
- Accounting del temps de connexió.
 - Utilitat ac:
>ac usuari
Per dia
>ac -d
 - Més informació:
http://www.gnu.org/software/acct/manual/html_mono/accounting.html

Accounting

- Accounting del sistema d'impressió
 - Sistemes BSD-lpd: Utilitzen la utilitat **pac**.
 - Habilitar el accounting d'impressió.
 - /etc/printcap

```
laser | postscript | ps1:\
```

```
:lp=/dev/lp:sd=/var/spool/ps1:if=/usr/local/sbin/filt:\
```

```
:mx#0:af=/var/adm/ps1_acct:pw#132:pl#66:pc#100
```

- Informes: Exemple:

```
pac -m -p0.12 -Psuper_impresora
```

| Login | pages | runs | price |
|-------|-------|------|-------------|
| david | 100.0 | 34 | USD 1200.00 |

Per cada usuari mostra les pàgines i el que s'ha gastat imprimint a la super_impresora, considerant que cada pàgina costa 0.12 cèntims.

Serveis de xarxa 1

Administració de Sistemes Operatius

Dept. Matemàtica Aplicada i Anàlisi,

Universitat de Barcelona

Index

- Introducció a les xarxes
- Configuració de la xarxa TCP/IP
- Serveis de xarxa
 - FTP
 - Telnet-SSH
 - NFS
 - Servidor DNS
 - Correu electrònic
 - Compartició de fitxers SMB
 - Altres servers
 - DHCP
 - SNMP
 - NTP

Introducció a les xarxes: Objectius

- Conèixer els conceptes bàsics a tenir en compte en una xarxa d'ordinadors.
- Aprendre les comandes i la localització dels fitxers de configuració dels diferents serveis de la xarxa.
- Saber configurar un nou client en una xarxa d'ordinadors.
- Saber configurar els diferents serveis d'un servidor.

Index

- Introducció a les xarxes
- **Configuració de la xarxa TCP/IP**
(Visió del client)

Introducció a les xarxes: Vocabulari

- **TCP/IP:** Conjunt de protocols i serveis que es fan servir per a interconnectar sistemes informàtics.
- **Host:** Ordinador individual en un sistema, caracteritzat per un nom i una adreça IP.
- **Sistema local:** Host on està connectat l'usuari.
- **Sistema remot:** Qualsevol host diferent a on s'executen els processos de l'usuari.

Introducció a les xarxes: Vocabulari

- **Client:** Sistema que demana la realització d'una tasca.
- **Servidor:** Sistema que realitza una tasca per a un client.

Introducció a les xarxes: Medi físic

- **Medi físic:** Cablejat que interconnecta els ordinadors: coaxial, parell trenat, fibra òptica.
- **Adaptador de xarxa:** Interfície entre l'ordinador i el medi físic.
 - Media Access Control (MAC): Identificador globalment **ÚNIC** de cada adaptador individual.
 - Normalment consta de 48bits (280 trillions de MACs):

00:00:4e:f4:7a:32

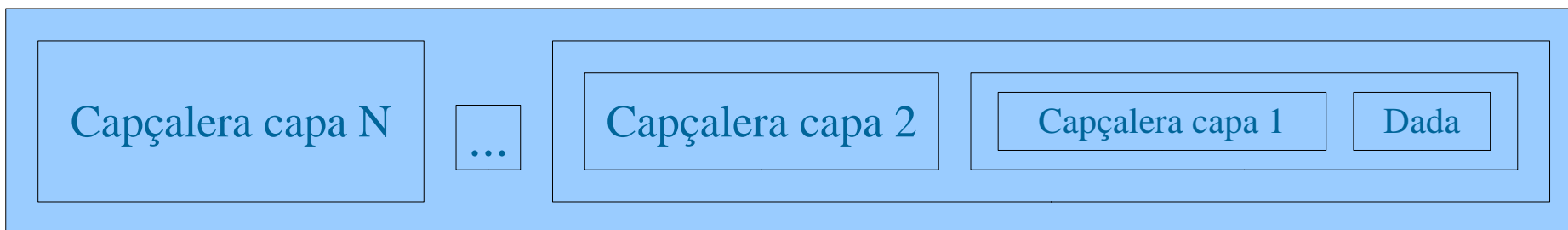
24 bits fabricant: Organizationally Unique Identifier (OUI)

Introducció a les xarxes: Medi físic

- Conèixer la MAC Address:
 - `ifconfig <interface>`
- **Interface:** Entitat lògica consistent en un adaptador de xarxa i la seva capa software associada al Sistema Operatiu.
- Interfícies d'un sistema:
 - **Loopback interface:** Permet al ordinador enviar-se missatges a si mateix. Esta implementada per Software.
 - **ethN:** Corresponent a la N-targeta de xarxa

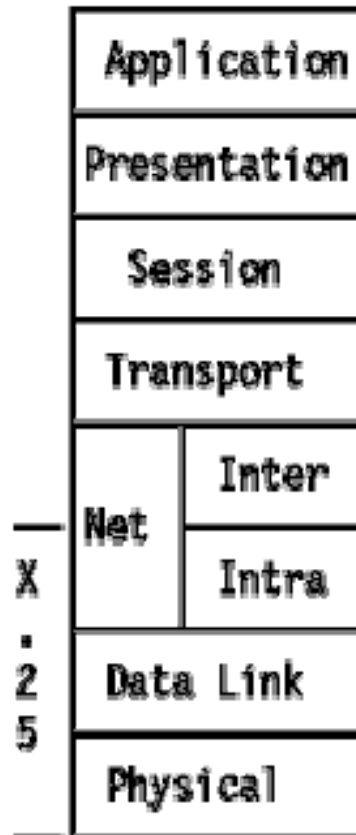
Configuració xarxa TCP/IP

- La comunicació per xarxa s'implementa en capes: model OSI.
- Cada missatge viatja d'una capa al seu nivell superior, on s'afegeix la informació de la capa.
- La pila es desfarà en l'equip receptor en ordre invers.
- Open Systems Interconnection és la base del protocol TCP/IP.

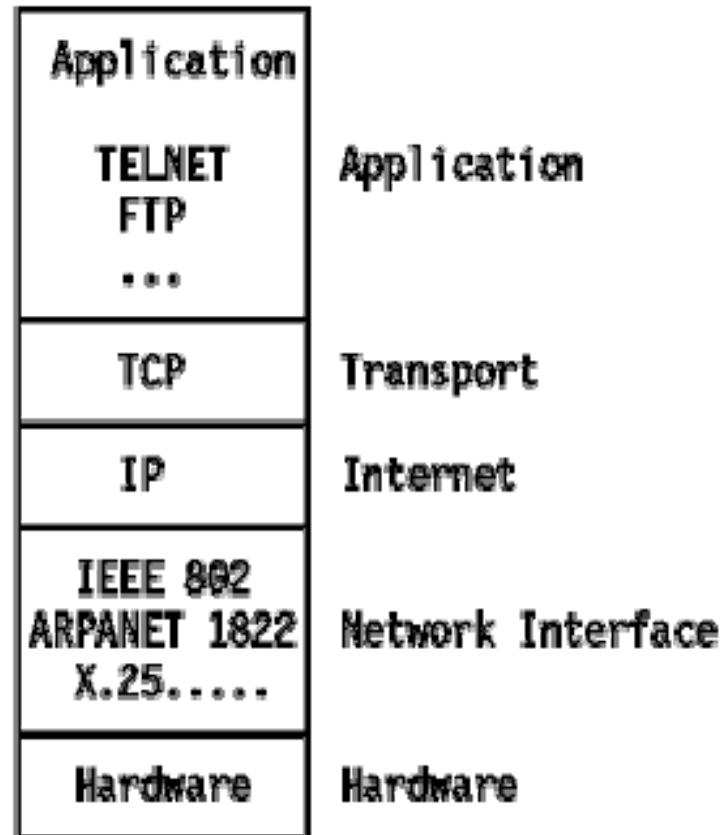


Configuració xarxa TCP/IP

ISO OSI Reference Model



TCP/IP protocols



Configuració xarxa TCP/IP

- El procés d'afegir la capçalera n-èssima a la dada que es rep de la capa inferior s'anomena **encapsulació**.
- El procés de dividir les dades en paquets més petits per a la seva transmissió s'anomena **fragmentació**.
- **MTU**: Unitat màxima de transmissió (1500bytes).
- **Serveis**: Facilitats del sistema per a desenvolupar una tasca. Ex: telnet, ssh, ftp,...
- **Dimonis (Daemons)**: Programa resident que ofereix algun tipus de servei (telnetd, sshd, ...).

Configuració xarxa TCP/IP

- Un servei està definit per:
 - Un **protocol** de transport.
 - Un **port**: Identificador lògic de l'aplicació.

```
/etc/protocols: Defineix el número de cada protocol  
/etc/services: Defineix els ports associats als  
serveis tcp i udp del sistema
```

- **Conexions TCP/IP: Estan determinades per un socket:**
 - IP màquina origen
 - Port a la màquina origen
 - IP màquina destí
 - Port de la màquina destí

Configuració xarxa TCP/IP

- Exemple:

- Un usuari logat a la màquina Veron (10.1.3.1) estableix una connexió per telnet a la màquina remota Ortega (10.1.3.5) pel port 23, usant com a port a la màquina d'origen el 2456.
- Qualsevol altra màquina podria connectar al port 23 del servidor (Ortega), ja que el socket es distingiria per la ip origen.
- De la mateixa manera es podria obrir una altra sessió de telnet desde Veron a Ortega, usant un port d'origen diferent.

Configuració xarxa TCP/IP

- Normalment els ports inferiors a 1024 estan reservats a serveis del sistema i estan restringits a root.
- Fitxer /etc/services conté una llista completa dels serveis del sistema i els ports associats.

Configuració xarxa TCP/IP

- Cada PC d'una xarxa rep un **hostname** únic associat a una única **adreça IP**.
- PC's amb múltiples adaptadors de xarxa poden tenir múltiples IP.
- Adreça de xarxa: Ipv4
 - Número de 32 bits normalment escrit com:
 $158.109.4.123/23$
 - Dividida en dues parts:
 - Bits més significatius: xarxa
 - Bits menys significatius: host dins la xarxa

$158.109.4.123/23$:
 $158.109.4.0 - 158.109.5.255$

Configuració xarxa TCP/IP

- Tipus de xarxes: adreça a.b.c.d

Rang Xarxa Host

- | | | | | | |
|----|-----------|---------|-----------|-------|------------------------------------|
| 0. | Classe A: | 1-126 | a | b.c.d | 126 xarxes de 16M host |
| 1. | Classe B: | 128-191 | a.b | c.d | 16K (6+8bits) xarxes de 65K hosts |
| 2. | Classe C: | 192-223 | a.b.c | d | 2M (5+8+8bits) xarxes de 254 hosts |
| 3. | Classe D: | 224-239 | Multicast | | |
| 4. | Classe E: | 240-254 | Recerca | | |

- Adreces especials:

- Host part = 0 \Rightarrow la pròpia xarxa

Ex: 0.0.0.0 fa referència a la pròpia xarxa local

- 127.0.0.1/8 es reserva per la loopback interface

- Adreça de Broadcast: Host name = 255

Ex: 10.255.255.255 per una de classe A

Configuració xarxa TCP/IP

- Les adreces les assigna una organització oficial.
- Les màquines d'una xarxa local que no estan directament connectades a internet tenen reservat un rang d'adreces:
 - 10.0.0.0 a 10.255.255.255
 - 172.16.0.0 a 172.31.255.255
 - 192.168.0.0 a 192.168.255.255
- **NAT** (Network Address Translation): S'encarrega de transformar les adreces privades en apropiades pel seu ús a internet.

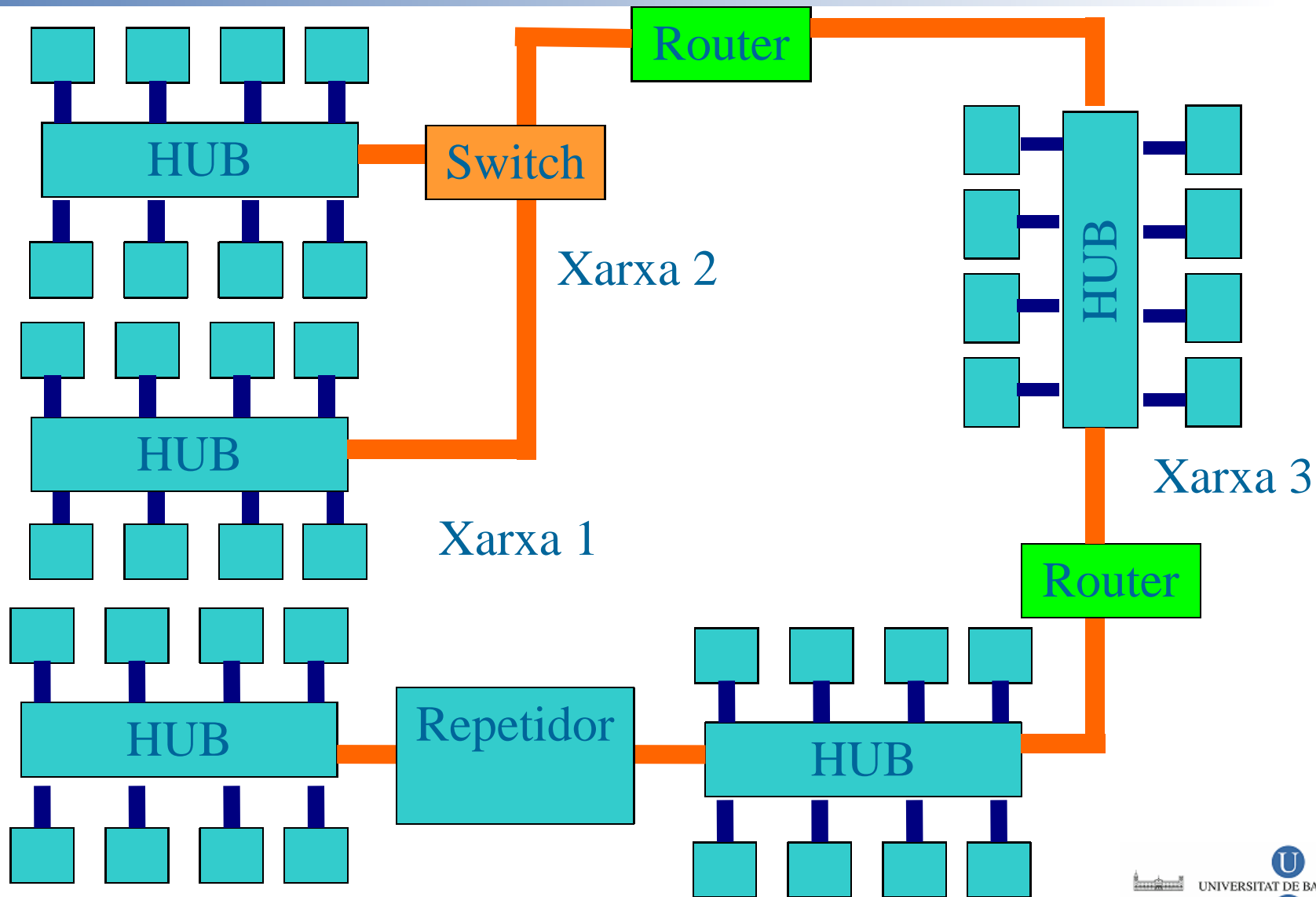
Configuració xarxa TCP/IP

- **Subnets:** Consisteixen en dividir els bits de l'adreça ip per a establir subxarxes en el format d'adreça.
 - Ex:
 - Una màscara de xarxa 255.255.255.192 (26) (11000000) per l'adreça de tipus C 192.168.10.0 indica que s'ha dividit la xarxa en 4 subxarxes de 62 hosts cadascuna.
- Les adreces IP son finites (2^{32}).
- Ipv6 és el nou protocol que defineix adreces de 128 bits compatible amb IPv4.

Configuració xarxa TCP/IP

- Dispositius d'interconnexió de màquines
 - **Hub**: Encarregat de enllaçar els diferents hosts d'una xarxa ethernet. (*Network layer*)
 - **Repetidor**: Enllaça dues xarxes que es troben separades per una longitud superior a la màxima permesa per l'ethernet. (*Network layer*)
 - **Switch**: Encadena dos segments de xarxa, evitant que dades d'un segment arribin al altre. (*Internet layer*)
 - **Bridge**: Switch que només té dos ports. (*Internet layer*)
 - **Router**: Dispositiu més sofisticat que interconnecta diferents xarxes i es capaç de definir rutes òptimes al destí. (*Transport layer*)

Configuració xarxa TCP/IP



Configuració xarxa TCP/IP

- Per introduir un nou host a la xarxa cal:
 1. Instal·lar el software associat al hardware de xarxa i comprovar que el kernel suporta la xarxa.
 2. Assignar un nom de host i adreça a màquina.
 3. Comprovar que els daemons associats als serveis s'executen al engegar la màquina.
 4. Configurar la resolució de noms.
 5. Definir el gateway per enviar paquets fora de la pròpia xarxa local.
 6. Efectuar un test bàsic de xarxa.

Configuració xarxa TCP/IP

1.-Configurar la interfície de xarxa

S'usa la comanda `ifconfig` (**interface config**)

```
ifconfig interface opcions adreces
```

Exemple 1:

```
ifconfig lo0 localhost up
```

Configura la interfície loopback (up indica activa, és la opció per defecte)

Exemple 2:

```
ifconfig eth0 inet 192.168.1.9 netmask 255.255.255.0
```

Configura la interfície de la primera targeta ethernet (eth0) assignant-li l'adreça ip 192.168.1.9 i la màscara de xarxa 255.255.255.0

Configuració xarxa TCP/IP

Es modifica el fitxer `/etc/network/interfaces`

Exemple:

```
iface eth0 inet static
    address 192.168.1.9
    netmask 255.255.255.0
    gateway 192.168.0.1
```

En el moment de l'arrencada s'executa el script

`/etc/rcS.d/S40networking`

Que s'encarrega d'activar totes les interfícies definides a
`/etc/network/interfaces`

Configuració xarxa TCP/IP

DHCP: Assignació de l'adreça IP de forma automàtica (Dynamic Host Configuration Protocol)

- Un servidor assigna una IP a un host per un període de temps que caduca.
- Útil per a entorns amb màquines mòbils (p.e. portàtils).
 1. El client realitza un broadcast al port 67 (usant com a adreça origen 0.0.0.0 i destí 255.255.255.255).
 2. Un o més servidors responen oferint una IP, una màscara i una caducitat.
 3. El client en tria una i fa un broadcast amb la seva selecció
 4. El servidor triat envia un ACK al client.
 5. Quan falta un 50% de temps per a caducar, el client intenta negociar la renovació de l'adreça.

Configuració xarxa TCP/IP

- Configuració d'un client DHCP

Cal que al fitxer

```
/etc/network/interfaces
```

Hi aparegui:

```
iface eth0 inet dhcp
```

Es necessari tenir instal·lats un dels següents paquets:

```
dhcp3-client
```

```
dhcpcd
```

```
pump
```

Configuració xarxa TCP/IP

2.-Assignar el nom a la màquina

S'executa un script al inici:

```
/etc/init.d/hostname.sh
```

que fixa el hostname que es troba a:

```
/etc/hostname
```

Configuració xarxa TCP/IP

3.-Comprovar que els daemons associats als serveis s'executen al engegar la màquina

Al script:

```
/etc/rcS.d/S40networking
```

es configura la red.

Consultar els daemons que estan corrent al sistema

Ex:

```
ps aux | grep sshd
```

Si sshd està associat a una interface no engegada per el script anterior no s'executa el dimoni.

Configuració xarxa TCP/IP

4.-Configurar la resolució de noms.

- Els humans treballem amb noms (host names).
- Els ordinador treballen amb números (adreces IP)

 Ens cal un mecanisme de traducció.

Hi ha dues opcions:

1.-Mantenir un fitxer local de traducció `/etc/hosts`

 Indicad per a xarxes petites no connectades a Internet.

 2.-Consultar un servidor DNS (Domain Name Server)

Indicat per a xarxes mitjanes/grans connectades a Internet.

Configuració xarxa TCP/IP

4.1.-Traducció d'adreces local

- El fitxer /etc/hosts conté els hosts coneguts de la xarxa

Ex:

```
127.0.0.1 localhost.localdomain localhost
```

```
10.1.2.4 veron.argentina.es veron
```

```
#altres hosts
```

```
10.1.3.5 Ortega
```

```
10.1.3.4 Herrera
```

```
10.1.3.3 Gallardo
```

```
10.1.2.1 LeCHuck
```

- Conté sempre un mínim de dues entrades: el loopback address i el nom del propi host tal i com és conegut a la xarxa.

Configuració xarxa TCP/IP

4.2.-Traducció d'adreces usant DNS

- Configuració del fitxer /etc/resolv.conf . Exemple:

```
search maia.ub.es
```

```
nameserver 192.168.9.44
```

```
nameserver 192.168.10.200
```

Search ens marca el domini on buscar el nom dels hosts

Nameserver serveix per establir per ordre les IP de les màquines que donen el servei DNS.

- Fitxer /etc/nsswitch.conf. Exemple:

```
host: files dns
```

Especifica l'ordre en que es consulten diferents serveis que fan la mateixa funció. En l'exemple, primer es miraria el /etc/hosts (files), i sinó es troba l'entrada es miraria el dns.

Configuració xarxa TCP/IP

5.-Definició del gateway.

- Quan s'ha d'enviar un paquet d'una màquina a una altra, cal definir quin camí ha de seguir el paquet.
- Depenent de les característiques de la xarxa trobem dos tipus d'encaminament (routing) de paquets:
 - Encaminament estàtic: Indicat per xarxes mitjanes-petites que no tenen molts camins redundants.
 - Encaminament dinàmic: Els camins que segueixen els paquets es determinen en temps de transmissió.

Configuració xarxa TCP/IP

5.1.-Encaminament estàtic

- Basat en la comanda **route**. Exemple:

```
route add -net 10.1.2.0 netmask 255.255.240.0 gw 10.1.3.100
```

Afegeix una ruta a la subxarxa 10.1.2 usant el gateway 10.1.3.100

Amb la comanda `netstat -r (n)` podem conèixer la actual taula d'encaminament. Exemple:

```
netstat -rn
```

| Destination | Gateway | Genmask | Flags | MSS | Window | irrt | Iface |
|-------------|---------------|---------------|-------|-----|--------|------|-------|
| localnet | * | 255.255.255.0 | U | 0 | 0 | 0 | eth0 |
| default | gw.maia.ub.es | 0.0.0.0 | UG | 0 | 0 | 0 | eth0 |

Encaminaria els paquets que van a fora de la xarxa a la màquina gw.maia.ub.es.

```
route del -net 10.1.2.0 netmask 255.255.240.0 gw 10.1.3.100
```

Elimina una línia de la taula d'encaminament.

Configuració xarxa TCP/IP

5.1.-Encaminament estàtic

- En debian es configura a través de `/usr/network/interfaces`. (Atenció: és diferent a la resta de distribucions!). Exemple:

Tenim una màquina amb dues interfícies:

`eth0:192.168.1.1`, i `eth1:192.168.2.1`

I volem afegir-hi dues rutes estàtiques:

- 1.Una cap a la xarxa `192.168.3.0` passant per la màquina `192.168.1.2` (Gateway).
- 2.Una cap a la màquina `192.168.4.23`, passant pel gateway `192.168.2.2`.

Definició en línia de comandes:

```
route add -net 192.168.3.0 netmask 255.255.255.0 gw 192.168.1.2
```

```
route add -host 192.168.4.23 gw 192.168.2.2
```

- Problema en apagar l'ordinador, perdem les rutes. Ens cal posar-les als scripts d'inici: `/etc/network/interfaces`

Configuració xarxa TCP/IP

5.1.-Encaminament estàtic

- Paraules clau: **up**, **down**. Permeten executar una comanda quan s'ha activat/desactivat una interfície. Exemple:

```
auto lo eth0 eth1
```

```
iface eth0 inet static
```

```
address 192.168.1.1
```

```
netmask 255.255.255.0
```

```
network 192.168.1.0
```

```
broadcast 192.168.1.255
```

```
up route add -net 192.168.3.0 netmask 255.255.255.0 gw 192.168.1.2
```

```
iface eth1 inet static
```

```
address 192.168.2.1
```

```
netmask 255.255.255.0
```

```
network 192.168.2.0
```

```
broadcast 192.168.2.255
```

```
up route -host 192.168.4.23 gw 192.168.2.2
```

Configuració xarxa TCP/IP

5.2.-Encaminament dinàmic

- Manté una taula d'encaminament variable en funció de l'estat actual de la xarxa.
- Dos tipus de protocols:
 - 1.- Distance-vector protocols: Es basen en assignar la ruta òptima a partir del nombre mínim de salts a seguir. Ex: RIP (Routing Information Protocol) .
 - 2.- Link-state protocols: Mantenen un mapa de la topologia de la xarxa. Requereixen molts més recursos, però tenen una capacitat d'adaptació més alta davant de caigudes de nodes. Ex: OSPF (Open Shortest Path First).

Configuració xarxa TCP/IP

5.2.-Encaminament dinàmic

- **RIP** (Routing Information Protocol): Cada router envia la seva taula d'encaminament a la resta, indicant el mínim número de salts que coneix per anar d'un a l'altre. Després cada router actualitza la seva informació.
 - Daemon **routed**: implementa el RIP en sistemes UNIX.
- **OSPF** (Open Shortest Path First): Cada router manté un graph dirigit modelant la seva xarxa. Aquestes dades es comparteixen amb els routers veïns que actualitzen els seus mapes.
 - Daemon **gated**: implementa el OSPF en sistemes UNIX.

Configuració xarxa TCP/IP

6.-Efectuar un test bàsic de xarxa

- Comprovació bàsica del funcionament del HW (LEDs)
- Comprovació de la connexió en xarxa:
 - ping Adreça_IP (`ping 161.116.83.1`)
- Comprovació del DNS
 - ping hostname (`ping www.maia.ub.es`)
 - nslookup (`nslookup www.maia.ub.es`)
- Comprovar els diferents serveis de xarxa: telnet, ...

Configuració xarxa TCP/IP

- Glossari de comandes d'utilitat:

hostname: Mostra el nom del host del sistema.

ifconfig: Configura les interfícies de xarxa.

ifconfig interfície: Mostra la configuració actual de la interfície que se li passa per paràmetre.

ifconfig -l llista les diferents interfícies.

ping: Envia un paquet de prova i mostra la resposta.

arp -a: mostra la taula de correspondències IP-MAC.

netstat: Mostra l'actual taula d'encaminament, estadístiques de les interfícies, etc.

route: Afegeix/elimina entrades a la taula d'encaminament.

traceroute: Mostra el camí que segueix un paquet des del host origen al destí. Ex: traceroute www.ub.edu.

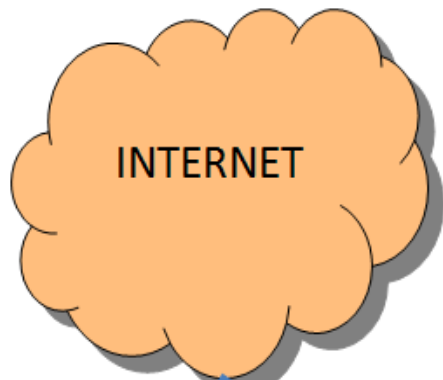
nslookup: Obté l'adreça IP a partir d'un nom de host.



UNIVERSITAT DE BARCELONA

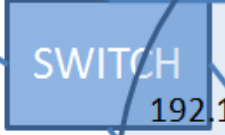
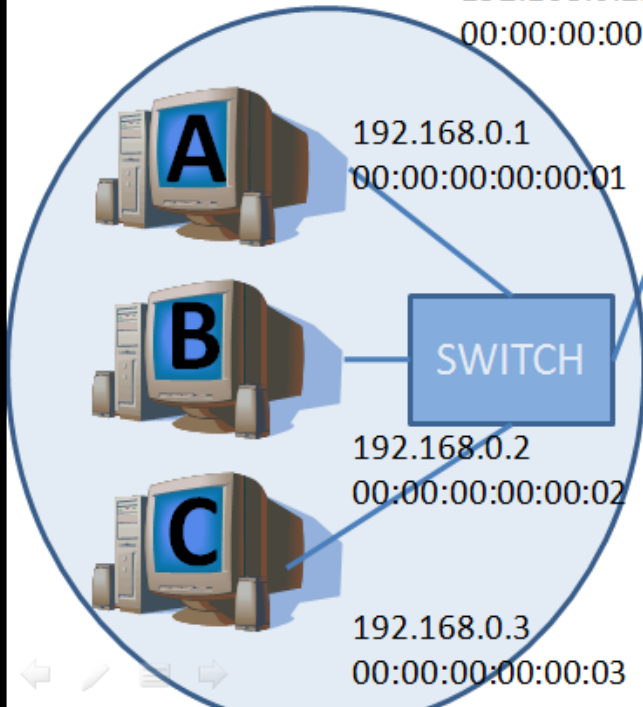


XARXA-3 192.168.3.0/24

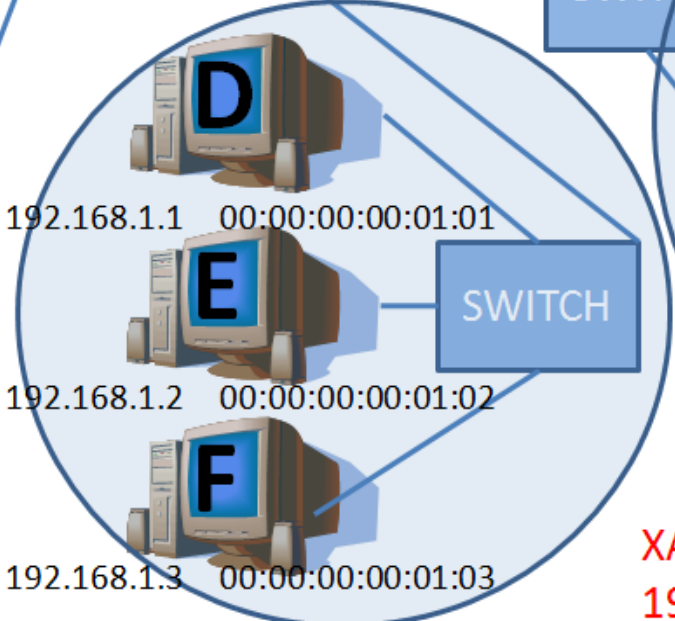


INTERNET

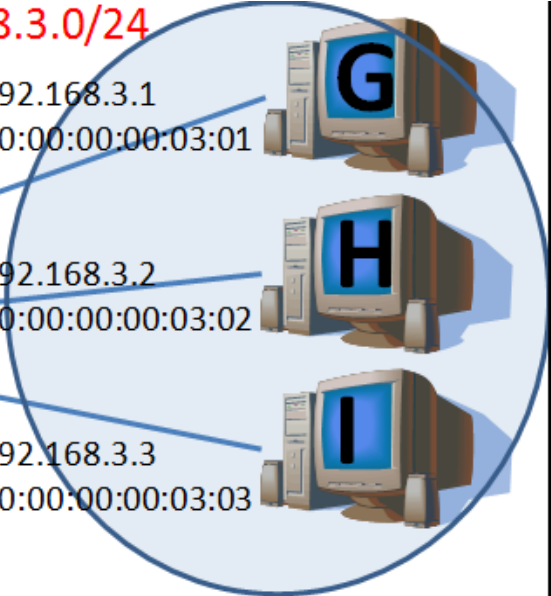
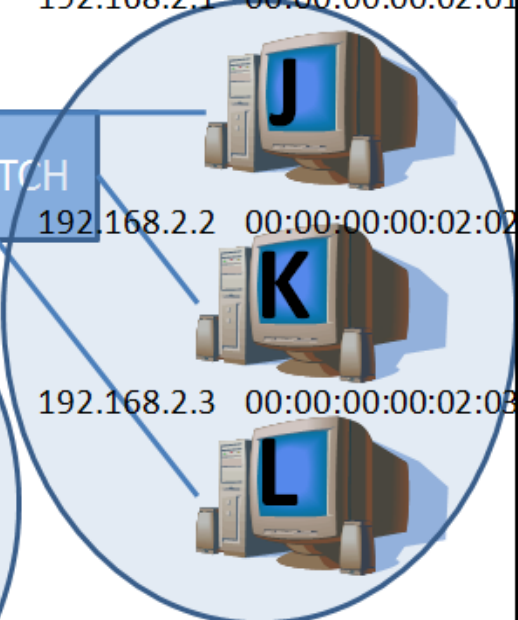
XARXA-0
192.168.0.0/24



XARXA-1
192.168.1.0/24



XARXA-2
192.168.2.0/24



1. Fer les taules d'encaminament dels routers 0 i 1 de tal forma que totes les xarxes es comuniquin amb totes.

Router0

| Destination IP | Gateway | Mask | iFace |
|----------------|---------|------|-------|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Router 1

| Destination IP | Gateway | Mask | iFace |
|----------------|---------|------|-------|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

2. Codifica les capçaleres internet i network dels paquets que van de la màquina H a la màquina A i com es modifiquen a cada router.

| | IP ORIGEN | MAC ORIGEN | IP DESTI | MAC DESTÍ |
|----------|-----------|------------|----------|-----------|
| SORTIDA | | | | |
| ROUTER-1 | | | | |
| ROUTER-0 | | | | |

1. Fer les taules d'encaminament dels routers 0 i 1 de tal forma que totes les xarxes es comuniquin amb totes.

Router0

| Destination IP | Gateway | Mask | iFace |
|--------------------|----------------------|----------------------|-------------|
| 192.168.0.0 | * | 255.255.255.0 | Eth0 |
| 192.168.1.0 | * | 255.255.255.0 | Eth1 |
| 192.168.2.0 | * | 255.255.255.0 | Eth2 |
| 192.168.3.0 | 192.168.2.253 | 255.255.255.0 | Eth2 |
| Default | 192.168.2.253 | 0.0.0.0 | Eth2 |
| | | | |
| | | | |

1. Fer les taules d'encaminament dels routers 0 i 1 de tal forma que totes les xarxes es comuniquin amb totes.

Router0

| Destination IP | Gateway | Mask | iFace |
|------------------------|--------------------------|--------------------------|-----------------|
| 192.168.0.0 | * | 255.255.255.0 | Eth0 |
| 192.168.1.0 | * | 255.255.255.0 | Eth1 |
| 192.168.2.0 | * | 255.255.255.0 | Eth2 |
| 192.168.3.0 | 192.168.2.253 | 255.255.255.0 | Eth2 |
| Default | 192.168.2.253 | 0.0.0.0 | Eth2 |
| | | | |
| | | | |

Router 1

| Destination IP | Gateway | Mask | iFace |
|----------------|---------------|---------------|-------|
| 192.168.3.0 | * | 255.255.255.0 | Eth1 |
| 192.168.2.0 | * | 255.255.255.0 | Eth0 |
| 192.168.0.0 | 192.168.2.254 | 255.255.255.0 | Eth0 |
| 192.168.1.0 | 192.168.2.254 | 255.255.255.0 | Eth0 |
| Default | 10.0.0.12 | 0.0.0.0 | Eth2 |
| | | | |
| | | | |

2. Codifica les capçaleres internet i network dels paquets que van de la màquina H a la màquina A i com es modifiquen a cada router.

| | IP ORIGEN | MAC ORIGEN | IP DESTI | MAC DESTÍ |
|-----------------|-------------|-------------------|-------------|-------------------|
| SORTIDA | 192.168.3.2 | 00:00:00:00:03:02 | 192.168.0.1 | 00:00:00:00:03:FE |
| ROUTER-1 | 192.168.3.2 | 00:00:00:00:02:FD | 192.168.0.1 | 00:00:00:00:02:FE |
| ROUTER-0 | 192.168.3.2 | 00:00:00:00:00:FE | 192.168.0.1 | 00:00:00:00:00:01 |

Serveis de xarxa 2

Administració de Sistemes Operatius
Dept. Matemàtica Aplicada i Anàlisi,
Universitat de Barcelona

Index

- Introducció a les xarxes
- Configuració de la xarxa TCP/IP
(Visió del client)

- **Serveis de xarxa**
(Visió del servidor)

- FTP
- Telnet-SSH
- NFS
- Servidor DNS
- Correu electrònic
- Compartició de fitxers SMB
- Altres servers
 - DHCP
 - SNMP
 - NTP

Serveis de xarxa: Introducció

- Objectius:
 - Conèixer els principals serveis de xarxa que pot oferir un servidor.
 - Aprendre les comandes i fitxers bàsics de configuració de cada servei.

Serveis de xarxa: FTP

- Protocol dissenyat per a la transferència de fitxers.
 - Una mica en desús a favor de Bittorrent i servidors http.
 - Existeix una gran quantitat de programes que implementen un servidor de ftp: ftpd, proftpd, etc.

Serveis de xarxa: FTP

Exemple: proftpd.

- Instal·lació: `apt-get install proftpd`
- Engregar / parar el servidor: `/etc/init.d/proftpd start/stop`
- Configuració bàsica: `proftpd.conf`
 - Limitar l'accés a una carpeta.
 - Permetre usuaris anònims.
- Més Informació:
<http://www.debian-administration.org/articles/228>

Serveis de xarxa: Telnet

- Protocol que permet la connexió remota a un terminal (**telephone network**). En desús.
 - Servidor: `telnetd`
 - Problemes de seguretat:
 - Molts daemons pateixen vulnerabilitats.
 - Dades NO encriptades a través de la xarxa, incloent nom d'usuari i password.
 - Manca d'un esquema d'autenticació: “Man-in-the-Middle attacks”.
- Implementacions: `telnetd`, `telnetd-ssl`, `krb5-telnetd`.

Serveis de xarxa: SSH

- Connexió a l'equip remot mitjançant un canal segur.
 - Utilitza criptografia de clau pública per autenticar a l'usuari.
 - Suporta **tunneling** d'aplicacions: Una aplicació client/servidor envia les dades a un client/servidor de SSH, que les envia al servidor/client de l'aplicació.
- Diferent servidors que implementen ssh: ssh-krb5, lsh-server, **openssh-server**.

Serveis de xarxa: SSH

- Exemple: **OpenSSH**

- Daemon sshd
- Utilitza com a fitxer de configuració:
`/etc/ssh/sshd_config`
- Manté les claus d'encryptació a `/etc/ssh`
- Activació de tunneling del sistema X:
`#X11Forwarding no`
`X11Forwarding yes`

Serveis de xarxa: NFS

- **Network File System:** Permet que diferents màquines d'una xarxa local puguin accedir a fitxers remots com si fossin locals (rfc1094).
 - Els usuaris no necessiten tenir un directori home a cada màquina, poden accedir per xarxa a una sola còpia del home al servidor.
 - S'aprofita molt més el hardware. Programes i dades usats per molts usuaris es poden guardar en un espai comú al servidor.

Serveis de xarxa: NFS

- Configuració d'un servidor:

- Instal·lar els paquets del servidor NFS:

```
apt-get install nfs-kernel-server nfs-common portmap
```

- Configuració de les carpetes que s'exporten, fitxer

```
/etc/exports. Exemple: /directory clients_autoritzats
```

```
/home 192.14.31.2(rw,no_root_squash) www.first.com(ro)
```

```
/usr 192.14.31.2/24(ro,insecure)
```

- Client: instal·lació: apt-get install nfs-common portmap

- Muntatge de les unitats

```
mount files.servidor.com:/home /mnt/nfs/home
```

- Muntatge automàtic en temps de boot: /etc/fstab

```
192.14.31.1:/home /home nfs rw,hard,nosuid 0 0
```

Servidor DNS

- Servei de traducció d'una adreça-hostname a una adreça IP.
 - Per xarxes grans el /etc/hosts no és suficient.
- DNS és una Base de Dades distribuïda repartida per Internet, organitzada jeràrquicament.
- Noms de domini: Estructurats en forma arborescent, concatenats per un punt '.'
 - maia.ub.es
- Top level domain (TLD): Etiqueta última (més a la dreta): .es, .org, .com, etc.

Servidor DNS

- Cada organització pot registrar el seu propi domini, i subdividir-lo en subdominis segons les seves necessitats.
- BIND (Berkeley Internet Name Domain): Servidor DNS utilitzat en UNIX.
- Una implementació de DNS ha de tenir:
 - Subrutines de llibreria per a resoldre adreces (**resolvers**)
 - Un daemon que fa les funcions de servidor de noms. (**named** en el cas de sistemes UNIX)
 - Zones d'autoritat: Conjunt de hosts dins d'un domini.
- Resolució inversa: Trobar el nom d'una màquina donada una IP.

Servidor DNS

- Tipus de servidors de noms:
- Depenent de com responguin a una petició de resolució, un servidor pot ser:
 - Recursiu: Donada una petició, si el servidor no la pot satisfer, la demana a un altre servidor remot.
 - Iteratiu: Quan no pot satisfer una petició, el servidor respon amb la referència a un altre servidor.
- Les respostes d'un servidor poden ser:
 - Autoritzades: Provenen de servidors que tenen una còpia oficial dels registres de la BD DNS.
 - No autoritzades: Provenen de servidors que han “après” una resolució d'adreça després de veure com es respon a una petició. Normalment guarden en una cache tant les peticions que s'han pogut resoldre com les que no.

Servidor DNS

- Els servidors de nom autoritzats poden ser:
 - **Servidors Master:** Tenen una còpia oficial dels registres de la BD per una zona, i no necessiten demanar informació a cap altre servidor sobre aquell domini. També anomenats primaris.
 - **Servidors slave** (esclaus): Obtenen la seva informació de servidors master. També anomenats secundaris.
 - **Servidors stub** (cap): Funcionen com els servidors esclaus però no manegen registres del propi host. S'usen per actualitzar servidors.
 - **Servidors ocults:** Són servidors funcionals que no s'han fet públics. Es poden fer servir només coneixent la seva IP.
- *Caching-only name servers:* Usen només la informació de traducció que retenen a la seva cache.
- *Forwarders:* No determinen directament l'adreça, sinó que serveixen per contactar amb altres servidors.

Servidor DNS

- Exemple: Configuració d'un servidor DNS en Debian: **BIND**.
- Obtenció dels paquets:`apt-get install bind bind-doc dnsutils`
- El fitxer `/etc/resolv.conf`:

```
domain elmeudomini.com
search elmeudomini.com
nameserver 127.0.0.1
nameserver x.x.x.x
```
- Li indiquem el nom del domini, i que es puguin fer cerques sense completar l'adreça (p.e buscar per "platon" en comptes de `platon.elmeudomini.com`).
- L'adreça del servidor de noms és la pròpia màquina (on es configurarà el BIND), i es pot definir un DNS secundari per si falla el local.

Servidor DNS

- Configuració del servidor de noms BIND: `/etc/named.conf` (`named.boot` en versions antigues)
- Format:

```
paraula_clau[argument]{  
    item; //comentaris  
    item; #comentaris  
}
```
- Per defecte configura el localhost i resolució inversa del loopback address:

```
zone "localhost" {  
    type master;  
    file "named.local";  
};
```

```
zone "127.in-addr.arpa" {  
    type master;  
    file "named.rev-local";  
};
```



Servidor DNS

- Exemple (extret de <http://www.debian.org/doc/manuals/network-administrator/ch-bind.html>):
- Per afegir un nou domini s'especifica amb la paraula clau `zone`, definint el fitxer on hi ha la informació de traducció. També cal definir el mecanisme de resolució inversa.

```
zone "foo.org" {  
    type master;  
    file "foo.db";  
};
```

```
zone "100.168.192.in-addr.arpa" {  
    type master;  
    file "192.168.100.db";  
};
```

- La resolució inversa fa un mapping de l'adreça ip pel nom del domini.
 - Compte: L'adreça IP està especificada en ordre invers (192.168.100.*)

Servidor DNS

- Creació dels fitxers de zona en /var/named (foo.db)

```
;
; BIND data file for foo.db
; /var/named/foo.db
;
@      IN      SOA      foo.org. root.foo.org. (
                        1998121401      ; Serial
                        604800           ; Refresh
                        86400            ; Retry
                        2419200          ; Expire
                        604800 )         ; Default TTL

      IN NS      dns.foo.org.

      IN MX     10  mail.foo.org.

www IN  A       192.168.100.10
news IN  A       192.168.100.20
mail IN  A       192.168.100.30
dns IN  A       192.168.100.40
dns2 IN  A       192.168.100.50
```



Servidor DNS

- Indica el nom de la zona, el seu responsable (root@foo.org), un número de sèrie (que cal augmentar cada cop que modifiquem el fitxer), el temps que es triga a actualitzar la base de dades, i altres paràmetres.
- Després s'indica quin és el servidor dns primari: **dns.foo.org**.

- En l'exemple s'observa com s'associa (host-ip)

www.foo.org → 192.168.100.10

news.foo.org → 192.168.100.20

mail.foo.org → 192.168.100.30

dns.foo.org → 192.168.100.40

dns2.foo.org → 192.168.100.50

- El fitxer de traducció inversa és similar variant l'ordre de l'entrada i la sortida.

Servidor DNS

- Creació dels fitxers de zona de traducció inversa (192.168.100.db)

```
;
; BIND reverse data file for 192.168.100.0
; /var/named/192.168.100.db
;
@      IN      SOA      foo.org. root.foo.org. (
                        1998121401      ; Serial
                        604800           ; Refresh
                        86400            ; Retry
                        2419200          ; Expire
                        604800 )         ; Default TTL

      IN      NS       dns.foo.org.

10    IN      PTR      www.foo.org.
20    IN      PTR      news.foo.org.
30    IN      PTR      mail.foo.org.
40    IN      PTR      dns.foo.org.
50    IN      PTR      dns2.foo.org.
```



Servidor DNS

- Afegir un servidor secundari:
- Cal afegir una entrada a named, especificant:
 - Que el servidor és esclau (type slave).
 - Que la informació de zona la obtindrà del master 192.168.100.40, el master (dns.foo.org).

```
zone "foo.org" {  
    type slave;  
    file "foo.db";  
    masters {  
        192.168.100.40;  
    };  
};
```

```
zone "100.168.192.in-addr.arpa" {  
    type slave;  
    file "192.168.100.db";  
    masters {  
        192.168.100.40;  
    };  
};
```

Servidor DNS

- Afegir un servidor secundari: modificacions als fitxers de zona:

```
;
; BIND data file for foo.db
; /var/named/foo.db
;
@      IN      SOA      foo.org. root.foo.org. (
                        1998121401      ; Serial
                        604800          ; Refresh
                        86400           ; Retry
                        2419200         ; Expire
                        604800 )        ; Default TTL

      IN NS      dns.foo.org.
IN NS dns2.foo.org.

      IN MX     10  mail.foo.org.

www IN  A      192.168.100.10
news IN  A      192.168.100.20
mail IN  A      192.168.100.30
dns IN  A      192.168.100.40
dns2 IN  A      192.168.100.50
```



Servidor DNS

- Canvis en els fitxers de zona.

```
i
; BIND reverse data file for 192.168.100.0
; /var/named/192.168.100.db
i
@      IN      SOA      foo.org. root.foo.org. (
                1998121401      ; Serial
                604800          ; Refresh
                86400           ; Retry
                2419200         ; Expire
                604800 )        ; Default TTL

      IN      NS       dns.foo.org.
IN NS dns2.foo.org. ; ADD THIS LINE ***

10 IN PTR www.foo.org.
20 IN PTR news.foo.org.
30 IN PTR mail.foo.org.
40 IN PTR dns.foo.org.
50 IN PTR dns2.foo.org.
```



Servidor DNS

- Indica que dns2.foo.org és un servidor autoritzat per a foo.org.
- Quan es produeixi un canvi als fitxers de zona, es notificarà a dns2.foo.org.

Servidor DNS

- Més informació:
 - “Essential System Administration”. Frisch A., O’Reilly & Associates, Inc. Pàgines 414-452.
 - DNS/BIND Ivan Moore.
 - Guia de administració de xarxes amb Linux. Olaf Kirch. Capítol 6.
 - Configuració Bind en Debian:

<http://www.debian.org/doc/manuals/network-administrator/ch-bind.html>

Servidor de Correu electrònic

- Introducció.
 - DNS MX Records
 - Alias
 - Forwarding
- Programes client
- Agents d'accés
- Agent de transport
- Recuperació de missatges
- Gestió del mailbox

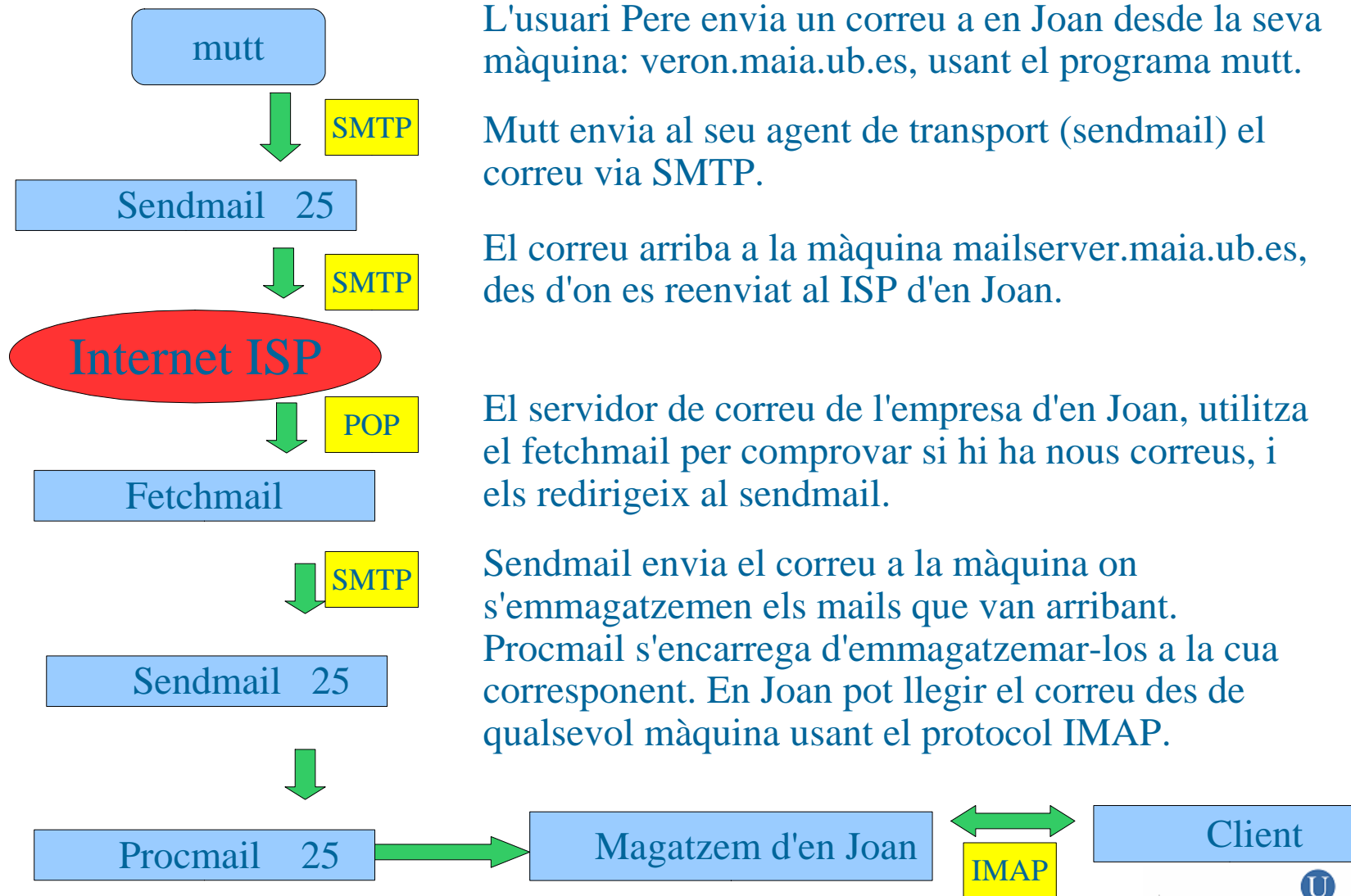
Servidor de Correu electrònic

- El correu és una eina fonamental pels usuaris, que depèn de molts processos que interaccionen:
 - **Programes client:** que permeten llegir a l'usuari els missatges emmagatzemats al servidor. Ex: mutt, pine, outlook, etc.
 - **Agents de transport de correu:** que s'encarreguen d'enviar els missatges a través de la xarxa cap a la màquina destí. Ex: sendmail. La majoria utilitzen el protocol SMTP (Simple Mail Transfer Protocol).
 - **Agents que reparteixen el correu** desde la màquina destí a la bústia de correu de cada usuari. Ex: procmail, mail, rmail, etc.

Servidor de Correu electrònic

- Programes que s'encarreguen de **recuperar missatges** (fetchmail) emmagatzemats al servidor **ISP**, que manté una cua de missatges que no han estat entregats degut a que la màquina destí no estava disponible.
- Programes que ofereixin **emmagatzemament del correu**, per a que l'usuari el pugui consultar desde diferents màquines (protocol IMAP, Internet Message Acces Protocol o POP, Post Office Protocol).

Servidor de Correu electrònic



L'usuari Pere envia un correu a en Joan desde la seva màquina: veron.maia.ub.es, usant el programa mutt.

Mutt envia al seu agent de transport (sendmail) el correu via SMTP.

El correu arriba a la màquina mailserver.maia.ub.es, des d'on es reenviat al ISP d'en Joan.

El servidor de correu de l'empresa d'en Joan, utilitza el fetchmail per comprovar si hi ha nous correus, i els redirigeix al sendmail.

Sendmail envia el correu a la màquina on s'emmagatzemen els mails que van arribant. Procmail s'encarrega d'emmagatzemar-los a la cua corresponent. En Joan pot llegir el correu des de qualsevol màquina usant el protocol IMAP.

Servidor de Correu electrònic

- Funcionalitats afegides al repartiment dels missatges:
 - Mecanisme per especificar el host associat a un correu (DNS MX Records).
 - A vegades pot interessar enviar el missatge a un host diferent al que estava definit.
 - Es pot fer un mapping d'adreces de correu amb comptes d'usuari (davidm@maia.ub.es a davidm@veron).
 - Sovint es fa necessari el Forwarding de missatges: redirigir el missatge a una adreça diferent (ex: quan un usuari canvia d'empresa,...).

Servidor de Correu electrònic

- DNS MX Records: Registres que permeten especificar cap a quina màquina es dirigiran els correus electrònics. Segueixen el format:

```
host [ttl] IN MX n desti
```

```
#ttl es un paràmetre opcional que indica la  
#caducitat del registre
```

On host es l'ordinador pel que es defineix el registre, n es un indicador de la prioritat del registre, i destí es l'ordinador on es redirigirà el correu. Exemple:

```
ortega:  IN MX 10 ortega  
        IN MX 20 aimar  
        IN MX 40 veron
```

especifica que la màquina ortega (persona@ortega) rep el seu propi correu, sinó està disponible es redirigirà a aimar i si aimar tampoc està disponible a veron.

Servidor de Correu electrònic

- Mecanisme d'**alias**: Consisteix en definir sobrenoms per a un usuari o grup d'usuaris (mailing lists).
- Exemple: `fitxer /etc/aliases`

format:

```
nom-alias: usuari, usuari, ...
```

```
dave: david_lopez@ignt.com
```

```
George: jordi_martin@appb.es
```

```
tots : dave, George
```

```
gent: :include:/ur/local/mail_list/llista_gent.list
```

Servidor de Correu electrònic

- **Forwarding** de missatges: Consisteix en reenviar el missatge a una altra adreça o destinació.
- Es pot fer a nivell de d'agent de transport (sendmail) ó a nivell d'usuari.
 - Forwarding definit per l'usuari: fitxer (a home dir):
`~/ .forward`

`david_lopez@novaempresa.com, "/home/davidl/mail_acumulat"`

Enviaria tots els mails que arribin a l'usuari david a l'adreça de correu especificada i en faria una còpia en un fitxer.

Servidor de Correu electrònic

- Consideracions i polítiques d'empresa en correu electrònic:
 - Minimitzar el nombre de destinataris (fer còpia Cc: només als usuaris necessaris).
 - Limitar l'ús particular del correu en l'organització (per propaganda comercial,...)
 - Limitar la mida dels attachments. Control de virus.
 - Advertir de la no privacitat de les dades (es reserva el dret de consultar el correu).
 - La notificació per correu electrònic no es considera legal en la major part dels casos.
 - Un correu es pot construir i falsificar, la informació que conté no es pot considerar verídica al 100%.

Correu electrònic: Programes client

- Existeixen multitud de clients de correu electrònic, els més comuns en els entorn UNIX son: **BSD mail**, **mutt** i **pine**. Els fitxers de configuració dels paràmetres per defecte de cada client són:
 - Mail: `/etc/mail.rc & ~/.mailrc`
 - Mutt: `/etc/Muttrc & ~/.muttrc`
 - Pine: `/usr/lib/pine.conf & ~/.pinerc`
- Enviament de missatges encriptats: PGP (Pretty Good Privacy). Exemple (mutt):

```
set pgp_decrypt_command="gpg --status-fd=2 %?p?--passphrase-fd 0?  
--no-verbose --quiet --batch --output - %f"
```

```
set pgp_sign_command="gpg --no-verbose --batch --quiet --output  
- %?p?--passphrase-fd 0? --armor --detach-sign --textmode %?a?-u  
%a? %f"
```

Correu electrònic: Agents d'accés

- L'administrador s'ha d'encarregar de:
 - Configurar un servidor per emmagatzemar el correu.
 - Configurar els daemons que gestionen el POP i l'IMAP al servidor.
 - POP: El client es connecta i descarrega els missatges al seu sistema local.
 - IMAP: El Client es pot descarregar els correus quan ho necessita, però es mantenen al servidor per a fer-los accessibles desde qualsevol client, distingint diferents estats en els missatges (llegit, no llegit, esborrat, ...).
 - Configurar els programes d'usuari per a que accedeixin al magatzem de correu en comptes de la seva bústia local.

Correu electrònic: Agents d'accés

- Daemons: `pop3d` i `imapd`
 - Normalment estan controlats per el "super servidor `inetd`", configurant les línies de `/etc/inetd.conf`:

```
pop3 stream tcp nowait root /usr/sbin/tcpd ipop3d
# imap stream tcp nowait root /usr/sbin/tcpd imapd
```

Indica que quan arribi una sol·licitud al port 110 (definit com a `pop3` a `/etc/services`) s'executara el programa `tcpd ipop3d`. En aquest cas `imap` està deshabilitat.
- La configuració a nivell de client depèn de l'aplicació de correu que es fa servir (`mutt`, `pine`, `netscape`, etc.). Ex: `mutt (a muttrc)`

```
set imap_user=david
```

```
set imap_pass=xxxxx #per evitar que preguntí a cada connexió
```

```
set imap_checkinterval = 900 #comprova cada 900 segons
```

```
set imap_passive=no
```

Correu electrònic: Agent de transport

- Rep els missatges dels gestors de correus, i els transporta per la xarxa fins a l'agent d'accés, que el posa a la bústia corresponent.
- Agents: **sendmail**, postfix, smail, qmail, exim.
- És el procés més complex de configurar per a un administrador. Hi ha versions comercials que permeten configuració via GUI.

Correu electrònic: Agent de transport

Sendmail

- Consta d'un daemon, un conjunt de comandes i diversos fitxers de configuració.

```
sendmail -bd q30m
```

Engega el sendmail daemon (que mira la cua cada 30 min.).

- Fitxer de configuració: sendmail.cf. Resulta complexe i difícil d'aprendre → **m4**, sistema de preprocés basat en macros.
- Passos:
 - Escollir (o crear) i modificar un dels fitxers de macros (*.mc).
 - Executar el processador de macros m4 per generar el nou sendmail.cf.
 - Copiar el fitxer al directori corresponent (/etc/sendmail.cf).
 - Notificar al daemon que ha de rellegir la configuració.

Correu electrònic: Agent de transport

Sendmail

- Sistema de macros m4. Consta de:
 - Invocació de macros existents: FEATURE('arguments','...',...).
 - Definició de noves macros: define('arguments','...',...).
 - Comentaris: entre divert(-1) i divert(0), o darrere de dnl (comenta una línia).
- Exemple:

```
divert(-1)
#####
# Sample configuration file for vstout
#####
divert(0)
VERSIONID(`@(#)sendmail.mc 8.7 (Linux) 3/5/96')
```

Especifica la versió del fitxer font

Correu electrònic: Agent de transport

Sendmail

```
OSTYPE(`linux')
```

OSTYPE. especifica el sistema operatiu.

```
FEATURE('smrsh')
```

```
define('PROCMAIL_MAILSERVER_PATH',`/usr/bin/procmail')
```

```
FEATURE(`local_procmail')
```

Defineix procmail com gestor per defecte.

```
define(`SMART_HOST',`veron.com.')
```

```
define('STATUS_FILE',`/var/log/mail.stats')
```

Especifica el concentrador de correu sortint fora del domini local, i el fitxer on es guarden les estadístiques.

```
MAILER(`smtp')
```

```
MAILER(`procmail')
```

Especifica smtp i procmail per repartir els missatges.

Correu electrònic: Agent de transport

Sendmail

Un cop definits els paràmetres de les macros cal generar el nou `sendmail.cf`:

```
m4 ../m4/cf.m4 test.mc > test.cf
```

Generació del nou fitxer.

```
cp /etc/mail/sendmail.cf /etc/mail/sendmail.cf.backup
```

```
cp test.cf /etc/mail/sendmail.cf
```

Guardar una còpia de seguretat.

```
chmod 444 /etc/mail/sendmail.cf
```

Canviar els permisos.

Correu electrònic: Agent de transport

Sendmail

Configuració del concentrador de correu:

```
FEATURE(`fitxer_de_hosts_locals')
```

Defineix un fitxer extern de configuració on hi ha la definició de hosts i dominis als que el sistema reconeix, envia el correu localment.

```
MASQUERADE_AS(`organitzacio.com')
```

```
FEATURE(`masquerade_envelope')
```

```
FEATURE(`allmasquerade')
```

Masquerade permet que tots els correus que surten del sistema sembli que surten del mateix usuari, eliminant les referències a la màquina local.

```
MAILER(`smtp')
```

```
MAILER(`local')
```

Especifica smtp i el gestor local com a agents per repartir els missatges

Correu electrònic: Agent de transport

Sendmail

Facilitats addicionals:

- Protecció i filtrat anti-spam.
- Ús de base de dades de "llista negra". Permeten rebutjar correus que venen d'un determinat usuari o organització.
- Monitorització de les operacions (utilitats: mailq, mailstats, hoststat).
- Ús d'àlias.

Servidor de Correu electrònic

Sendmail

- Més informació:
 - “Sendmail, Brian Costales”, Eric Allman, O’Reilly & Associates, Inc.
 - “Essential System Administration”. Frisch A., O’Reilly & Associates, Inc. Pàgines 542-578.
 - www.sendmail.org

Correu electrònic: Recuperació de missatges

- **Fetchmail:** Programa que permet la connexió automàtica a un servidor de correu remot per enviar els missatges al port 25 (SMTP) de la màquina local, de manera que l'agent de transport els veu com un missatge entrant més.

```
fetchmail -d 900
```

- Engega el daemon que cada 900 segons mira al servidor per veure si hi ha correu.

```
fetchmail servidor.aol.com
```

- Mira un cop directament al servidor sense esperar.
- Fitxer de configuració: `~/fetchmailrc`, permet especificar el número de segons, fitxers de log, servidor remot, etc.

Correu electrònic: Gestió del mailbox

- Agents que afegeixen al sistema una sèrie de facilitats com:
 - Llegir, i ordenar els missatges de la bústia segons els criteris definits per l'usuari (data, subject, usuari, etc.).
 - Rebutjar missatges d'usuaris específics.
 - Identificar i filtrar missatges de spam.
 - Comprovar la presència de virus i/o macros en attachments.
- **Procmail:** Es pot fer servir com:
 - Agent local repartidor de correu, rebent directament els missatges de l'agent de transport.
 - Fent un pipe del mail que arriba a l'usuari al procés procmail. Això s'especifica en el `~/ .forward`

```
"IFS=' '" && exec /usr/bin/procmail -Yf- || exit 75 #username"
```

Correu electrònic: Gestió del mailbox

- Configuració de procmail:
 - Utilitza el fitxer global `/etc/procmailrc` i el local `~/procmailrc`.
 - Es basa en definir un conjunt de regles que redirigeixen el correu en funció d'una condició.

Format:

```
:0[flags] #l'inici
          d'una regla
*condicio #expressió
          regular
desti del missatge
```

Exemple:

```
:0
* ^From.*jerk@bad-guys.org
/dev/null

:0c:
archive
```

Enviaria tots els missatges provinents del usuari jerk de bad-guys.org a `/dev/null`. Sinó els còpia a l'arxiu de correu.

Correu electrònic: Gestió del mailbox

- Configuració de procmail:

- Existeixen multitud d'opcions: forwarding, ordenació, pipes, execució de scripts (shell, perl)...

<http://lipas.uwasa.fi/~ts/info/proctips.html>

- Es pot usar com a filtre de spam, fent servir regles com:

```
:0
```

```
* !To .*sergio
```

```
/dev/null
```

```
:OH
```

```
* cdw buyer|sex|gift|job oportunitat|cialis|win
```

```
/dev/null
```

la primera descarta tots els correus que no vagin dirigits directament a mí, i la segona filtra per les paraules que són característiques dels mails de spam.

Correu electrònic: Gestió del mailbox

- Configuració de procmail:
 - Procmail com a virus scanner: permet l'execució de scripts en les regles:

```
:0fw
```

```
| /usr/local/bin/sanitizer.pl /etc/sanitizer.cfg
```

executa un script que analitza els virus segons un fitxer de configuració.

<http://mailtools.anomy.net/sanitizer.html>

Servidor de Correu electrònic

- Més informació:
 - “Essential System Administration”. Frisch A., O’Reilly & Associates, Inc. Pàgines 521-615.
 - <http://www.procmail.org/>
 - <http://www.sendmail.org/>
 - <http://www.fetchmail.info/fetchmail-man.html>

Compartició de fitxers

- Samba-SMB(**S**erver **M**essage **B**lock):
Implementació lliure del protocol SMB característic dels entorns MS-Windows i OS/2.
- Permet entre d'altres:
 - Compartir fitxers en un servidor UNIX, sent visibles desde hosts Windows.
 - Accedir a carpetes Windows.
 - Compartir impressores.

Compartició de fitxers: Samba

- Samba: Instal·lació: `apt-get install samba samba-client`

- Configuració de samba `/etc/samba/smb.conf`:

- Dividit en seccions, una per servei. Secció global:

```
[global]
workgroup = WGP
netbios name = DEBIANSERVERNAME
server string = %h server (Samba %v)
log file = /var/log/samba/log.%m
max log size = 1000
syslog = 0
```

- Compartir la carpeta `/my_shared_folder`

```
[share2]
path = /my_shared_folder
comment = Some random files
```

Compartició de fitxers: Samba

- Samba daemons: **nmbd, smb, winbind**.
- Test del fitxer de configuració:
`testparm /etc/samba/smb.conf`
- Re-engegar el servidor per llegir la configuració:
`/etc/init.d/samba restart`
- Mes informació sobre la instal·lació i configuració de samba:

<http://us5.samba.org/samba/docs/man/Samba-HOWTO-Collection/>

Altres Serveis

- Servidor DHCP
- SNMP
- NTP: Sincronització temporal

Altres Serveis: DHCP Server

- Donada una llista d'adreces IP, el servidor DHCP s'encarrega de proporcionar-ne una a cada sol·licitant.
 - Algunes adreces poden estar excloses, per exemple, perquè s'han assignat de forma fixa a una màquina.
 - Pot haver-hi una assignació reservada d'adreces a hosts específics, caracteritzats per la seva MAC address.
- S'usa el daemon dhcpd i el fitxer de configuració /etc/dhcpd.conf.
- Existeix un DHCP relay que s'encarrega de substituir al dhcp server quan aquest ha caigut.

Altres Serveis: DHCP Server

- Exemple de fitxer de configuració `dhcpd.conf`

```
subnet 192.168.0.0 netmask 255.255.255.0 {  
# --- default gateway  
option routers          192.168.0.1;  
option subnet-mask      255.255.255.0;  
option domain-name      "domain.org";  
range dynamic-bootp 192.168.0.128 192.168.0.255;  
default-lease-time 21600;  
max-lease-time 43200;  
host especial {  
hardware ethernet 12:34:56:78:AB:CD;  
fixed-address 207.175.42.254;  
}  
}
```


Altres Serveis: SNMP

- Existeix un conjunt d'eines que permeten examinar i controlar el tràfic d'una xarxa complexa.
- Segueixen el SNMP (Simple Network Management Protocol).
 - Protocol dissenyat per recollir dades de dispositius diversos (switch, routers, hosts,...).
 - Existeix una màquina, la NMS (Network Management Station) que monitoritza la informació de tots els dispositius.
 - La NMS pot enviar periòdicament peticions d'informació als dispositius.
 - Els dispositius (agents) poden enviar interrupcions a la NMS (manager), quan algun paràmetre està fora dels rangs acceptables.

Altres Serveis: SNMP

- Cal mantenir una convenció sobre els noms i valors que els agents poden passar al manager. Aquest camps estan definits al que s'anomena **MIB**: Management Information Bases.
- El paquet Net-SNMP conté les eïnes que fan la funcionalitat descrita per entorns UNIX/LINUX:
 - Snmptranslate: Proporciona informació sobre la estructura MIB.
 - Snmpget: Obté informació sobre algun dels camps del MIB.
 - Snmpset: Modifica la informació d'algun dels camps del MIB.
 - Snmptrap: Genera interrupcions quan cal avisar al manager sobre una situació d'excepció.
 - Snmpd: Daemon resident del SNMP (/usr/local/share/snmp/snmpd.conf)
- Més informació:
 - <http://www.net-snmp.org/>
 - Essential SNMP, Douglas Mauro i Kevin Schmidt (O'Reilly & Associates)

Altres Serveis: NTP

- Sovint els processos de xarxa depenen de la bona sincronització i del coneixement exacte del temps per a funcionar (time-outs, etc.).
- NTP: Network Time Protocol està dissenyat per a solucionar els problemes de sincronització a la xarxa.
 - Funciona com un sistema jeràrquic de servidors, que es connecten remotament a un rellotge extern atòmic, satèl·lit, NIST (National Institute of Standards and Technology).
 - El client ajusta periòdicament el seu rellotge basant-se el la informació rebuda del servidor, si hi ha més de 128 milisegons de diferència, es fa la correcció.
 - A `/etc/ntp.conf` s'especifica el servidor:

```
Server 192.168.15.33
Logfile /var/log/ntp
Driftfile /etc/ntp.drift
```

Bibliografia

- **Bibliografia recomanada:**

- Guia de administració de xarxes, Olaf Kirch.
<http://es.tldp.org/htmls/manuales.html>

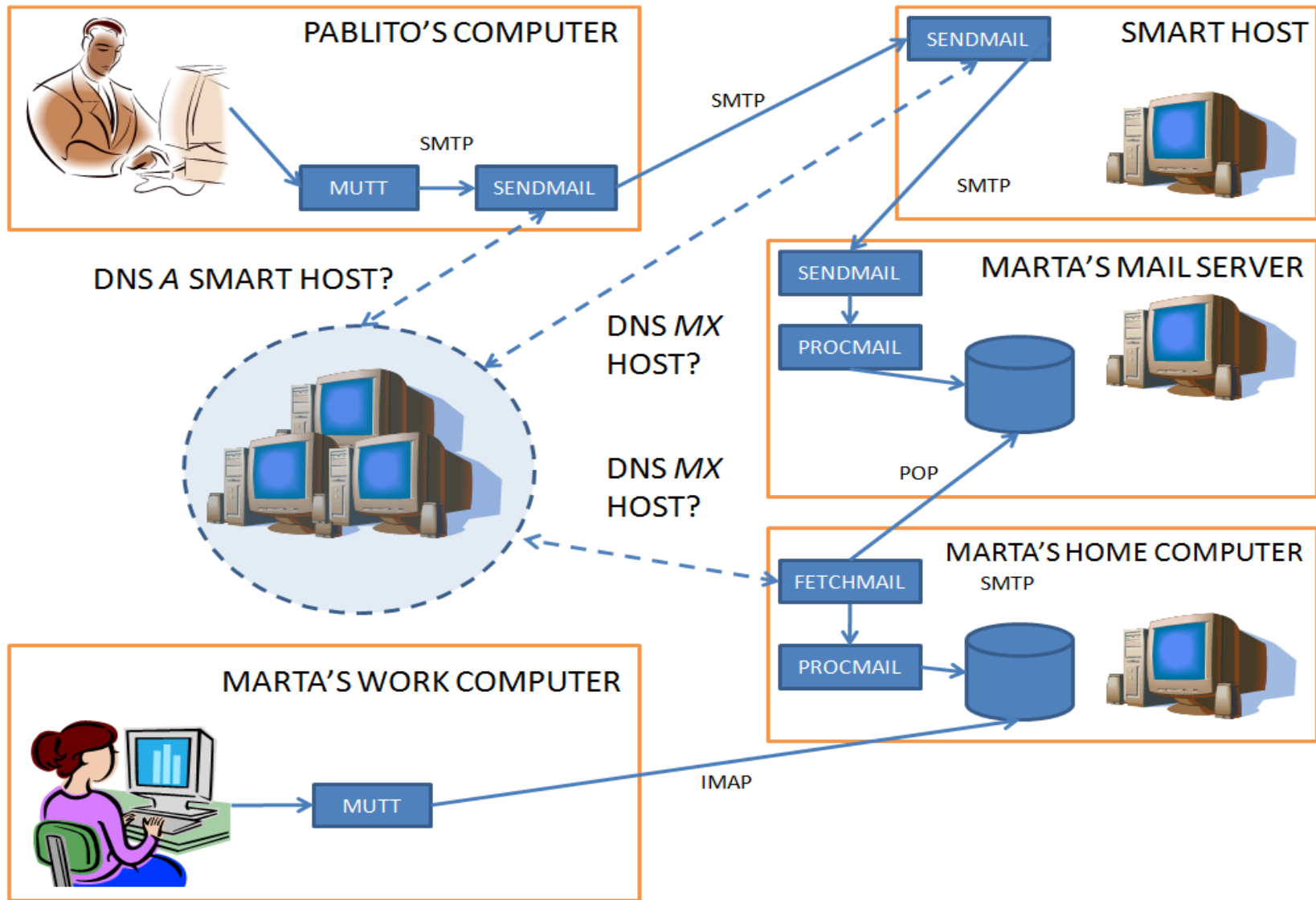
- Instal·lació en debian de serveis de xarxa:
<http://www.debianhelp.co.uk/>

- “Essential System Administration”.

Frisch A., O’Reilly & Associates, Inc.

- Documentació online debian, Network configuration.
<http://www.debian.org/doc/manuals/reference/ch-gateway.en.html>

Exemple correu més resolucions DNS



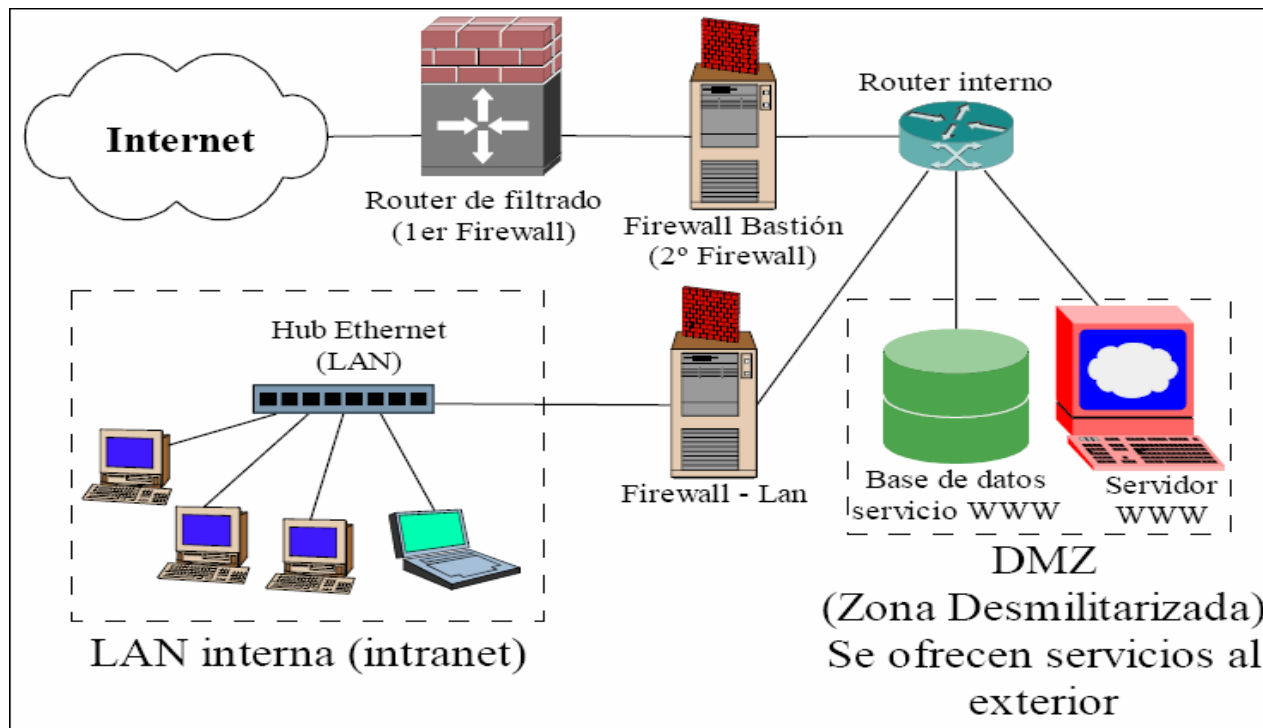
Firewall

Administració de Sistemes Operatius
Dept. Matemàtica Aplicada i Anàlisi,
Universitat de Barcelona

Seguretat en xarxa

- **Firewalls - Tallafocs**

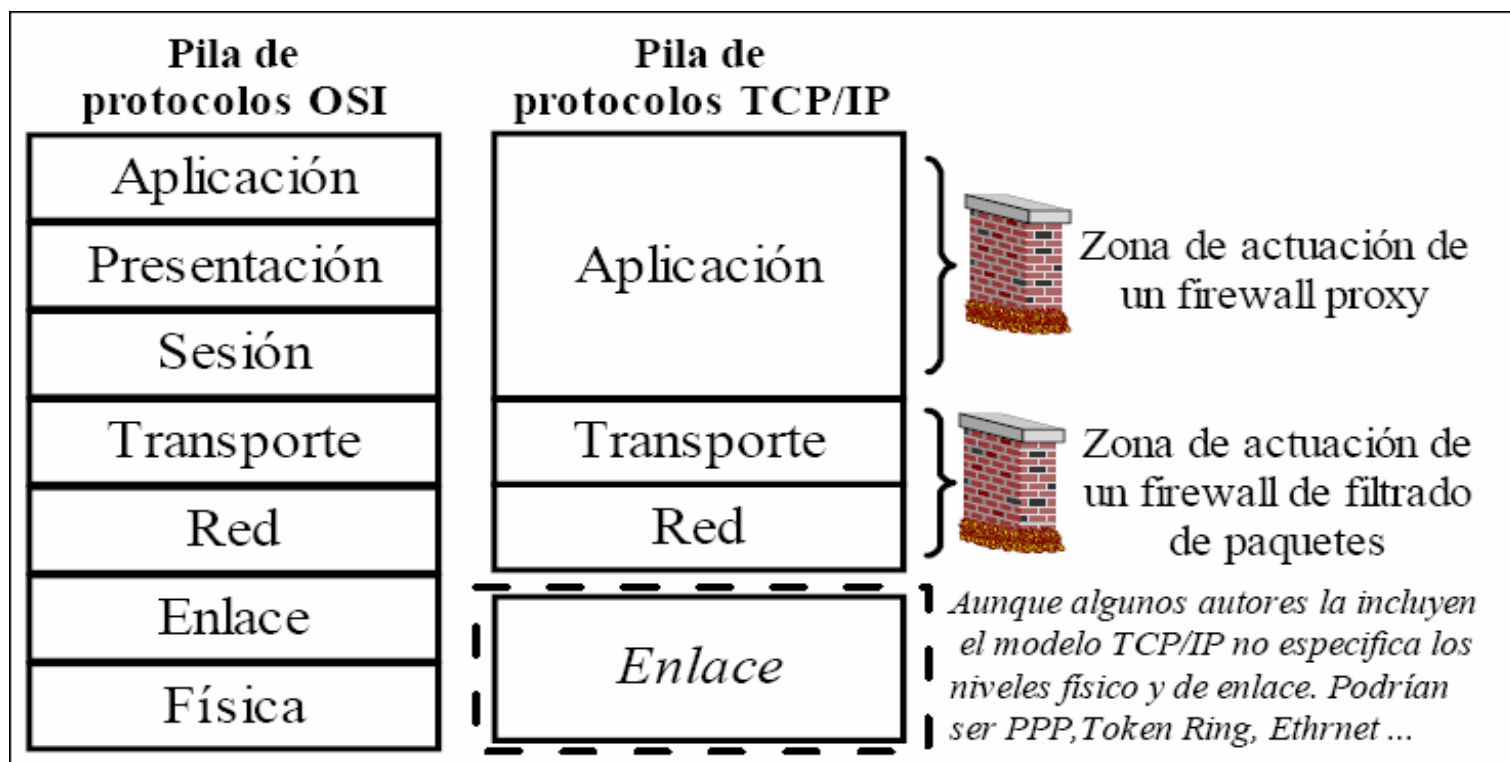
- Situat entre 2 o més xarxes emprant una directiva de seguretat entre les comunicacions.



Seguretat en xarxa

- **Firewalls**

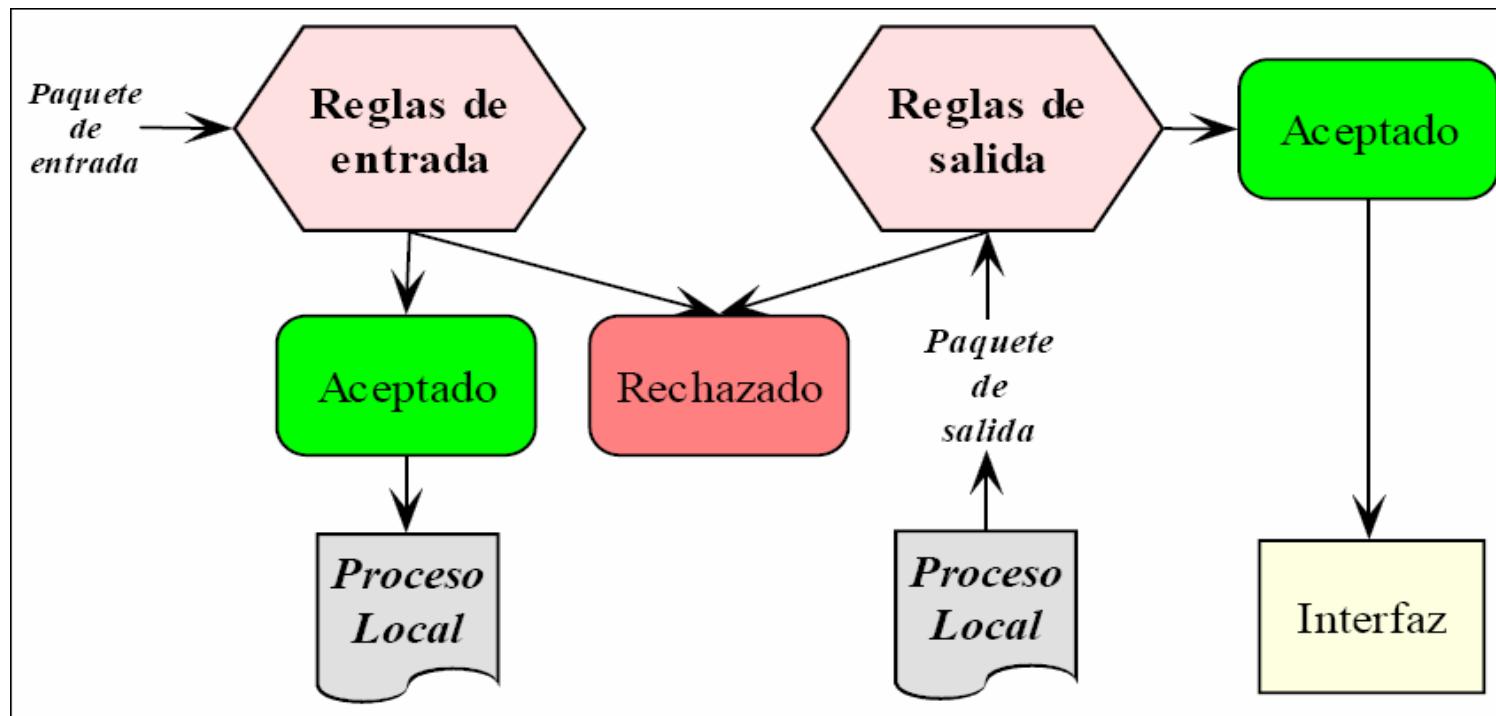
- 2 nivells de firewalls: a nivell d'aplicació i de xarxa i transport.



Seguretat en xarxa

- **Firewalls**

- A nivell de transport i xarxa TCP/IP IPFW.
- Filtra comunicacions internes i externes.



Seguretat en xarxa

- **Firewalls**
- Cadenes Regles Tipus
 - Acceptar.
 - Rebutjar: sense confirmació.
 - Denegar: confirmació.
- Comodins o màscares per defecte, 2 polítiques:
 - Acceptar tot i especificar les negacions.
 - Rebutjar tot i especificar les acceptacions (utilitzada).

Seguretat en xarxa

- **Firewalls**
- Filtratge **IPWF**: no sols capçalera IP, també els protocols TCP i UDP basant-nos en direcció origen, destí, port origen i destí, bits del TCP, etc.
- Regles generals:
 - *Filtratge de direcció remota*: Disminuir possibilitats d'atacs *ip spoofing*. (suplantar-se per una altra IP), podem fer-ho denegant:
 - IP pròpia de xarxa externa.
 - Direccions IP privades (tipus A (10.0.0.0 fins 10.255. 255. 255), B (172.16.0.0 fins 172.31. 255. 255) i C (192.168.0.0 fins 192.168. 255. 255)).
 - Direccions IP de multi-difusió (tipus D 224.0.0.0 fins 239. 255. 255. 255).
 - Direccions reservades de tipus E (240.0.0.0 fins 247.255.255.255).
 - Direccions de loopback (127.0.0.0 fins 127.255.255.255)

Seguretat en xarxa

- **Firewalls**

- Filtratge de direcció destí local: per defecte la interfase de xarxa ignora els paquets que no es dirigeixin cap a ella.
- Filtratge de port origen remot i destí local: configurats en funció dels serveis de xarxa.
- Filtratge de l'estat de la connexió TCP: exigir de paquets remots bit AWK activat.
- Sondejos i exploracions de ports: monitorització.

Seguretat en xarxa

- **Firewalls**

- Control·lar les activitats de la xarxa.
- Control de l'enviament de paquets.
- Ajustar l'ampla de banda.
- No requereix grans recursos.
- Adaptar a les necessitats de la xarxa.
- La majoria d'atacs provenen de la xarxa interna.
- Utilitzar NetFilter (<http://www.netfilter.org>), IPTables, FWBuilder (software gràfic).

Seguretat en xarxa

- Kernel 1.1 ipfw
- Kernel 2.0 ipfwadm
- Kernel 2.2 ipchains
- Kernel 2.4 iptables

IPTABLES mòdul NETFILTER:

- Tallafocs.
- NAT.
- Manipulació MANGLING.

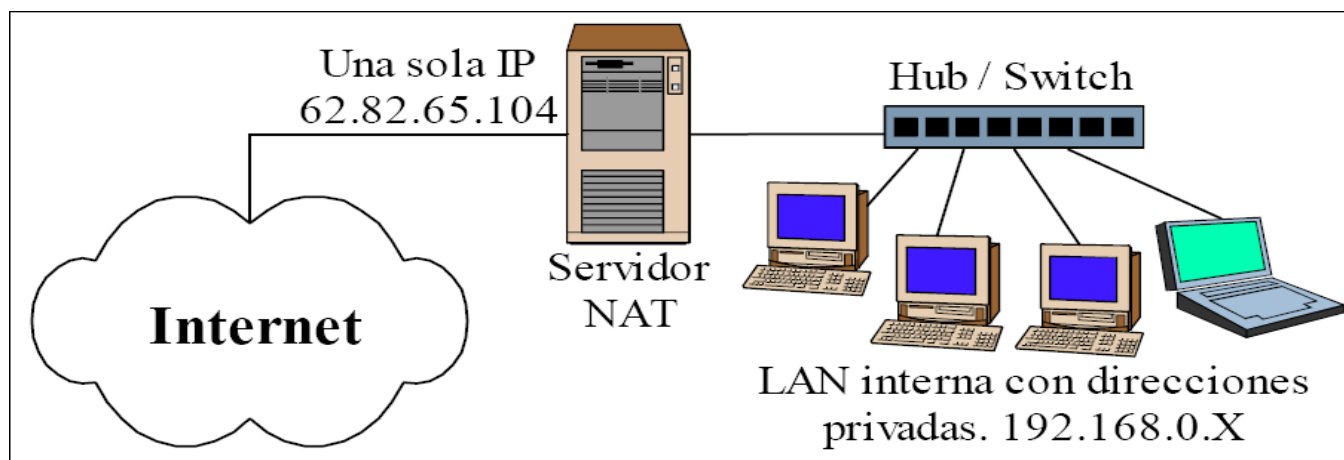
Seguretat en xarxa

- **IPTABLES - NAT**

- NAT: Network Address Translation

- Emmascarament o masquerading :

- Connexió de diferents màquines amb la mateixa direcció IP.



- Balanceig de càrrega: la IP d'una màquina servidor se li assignaria a una màquina que actuaria de **radware** (balancejador) i repartiria la càrrega entre diferents màquines.

Seguretat en xarxa

- **IPTABLES**

- Passos generals:

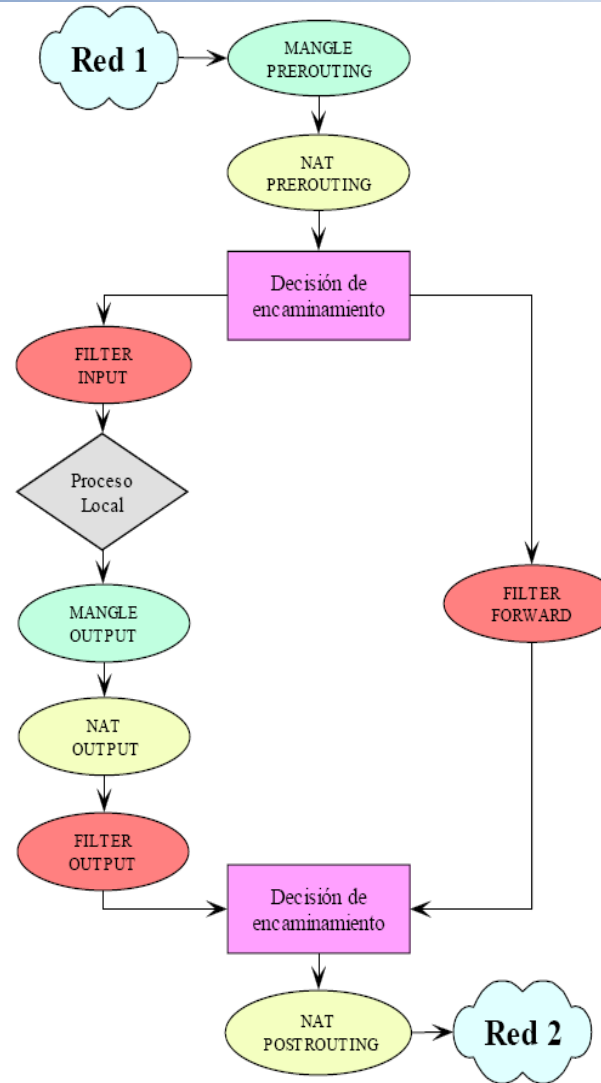
- Es crida el tipus de taula.
 - S'usen chains/cadenes definides per taula.
 - S'usen les regles de filtratge de les cadenes.
(condició/acció sobre atributs IP).
- Recorregut seqüencial, s'executa la primera coincidència, sinó la per defecte.
- Accions generals: ACCEPT, REJECT i DROP.

Seguretat en xarxa

- **IPTABLES**
- Iptables -t taula:
 - Nat:
 - PREROUTING: alterar paquets quan arriben al tallafocs.
 - OUTPUT: alterar paquets generats localment abans d'enrutar.
 - POSTROUTING: alterar paquets que acaben d'abandonar el tallafocs a l'origen.
 - Mangle:
 - Permet manipular altres elements dels paquets, com la TTL, etc. (PREROUTING, OUTPUT).
 - Filter:
 - INPUT: a paquets entrants al propi tallafocs.
 - FORWARD: paquets que arriben al tallafocs amb destí un altre host.
 - OUTPUT: paquets generats al propi host amb destins externs.

Seguretat en xarxa

- IPTABLES



Procesado de paquetes en netfilter-iptables

498

Seguretat en xarxa

- **IPTABLES**
- Crida general: iptables [-t taula] comanda [match] [objectius/salts]
 - (Emprar man per opcions detallades).
- **Opcions generals: afegir i manegar chains/cadenes:**

Iptables -N: crea una nova chain o capçalera buida (sense regles).

Iptables -X: elimina una chain buida.

Iptables -F: elimina totes les regles d'una chain.

Iptables -P: canvia la política per defecte d'una chain.

Iptables -L: llista les regles d'una chain.

Iptables -Z: posa a zero les variables d'auditoria d'una chain (número de paquets, de bytes, etc.).

Seguretat en xarxa

- **IPTABLES**
- Opcions generals: manegar les regles de les chains:

Iptables -A: afegir una nova regla al final d'una cadena.

Iptables -I: afegir una regla al començament d'una chain.

Iptables -R: reemplaça una regla d'una chain.

Iptables -D: elimina una regla d'una chain.

Seguretat en xarxa

- **IPTABLES**
- **Opcions generals: establir condicions i accions sobre cada regla (condicions de match entre paquets i regles):**

Iptables -s: indica un domini o IP (rang IPs) d'origen per avaluar les regles.

Iptables -d: indica un domini o IP (rang IPs) de destí per avaluar les regles.

Iptables -i (--in-interface): indica la interfase d'entrada per avaluar les regles.

Iptables -o (--out-interface): indica la interfase de sortida per avaluar les regles.

Seguretat en xarxa

- **IPTABLES**
- **Opcions generals: establir condicions i accions sobre cada regla (condicions de match entre paquets i regles):**

Iptables -p: especifica protocol del datagrama per aplicar la regla (nom del protocol o número):

TCP:

--sport: port d'origen del datagrama per concordar amb la regla.

--dport: port destí del datagrama per concordar amb la regla.

--tcp-flags: màscara per veure concordança de flags (SYN, ACK, FIN, RST, URG, PSH, ALL o NONE, separats per comes).

--syn: per coincidir amb datagrames amb SYN=1 i ACK=0 FIN=0.

UDP:

--sport: port d'origen del datagrama per concordar amb la regla.

--dport: port destí del datagrama per concordar amb la regla.

ICMP:

--icmp-type: identificar el tipus de missatge (té llista d'identificadors).

MAC:

--mac-source: direcció MAC (ethernet) per cadena INPUT i FORWARD.

Seguretat en xarxa

- **IPTABLES**
- **Opcions generals: establir condicions i accions sobre cada regla (condicions de match entre paquets i regles):**

Iptables -f: especificar accions sobre el segon i restants fragments del datagrama fragmentat

Iptables !: inverteix el valor lògic de la condició de la regla

Iptables -j: acció a fer si hi ha match a la regla

Seguretat en xarxa

- **IPTABLES**. *Exemple:*
- *Disseny d'un firewall senzill: Tenim un servidor web i ftp que volem fer públics per a tothom, i un servidor mysql que només volem que doni accés al nostre amic X.X.X.X perquè ens ajudi a gestionar la nostra base de dades.*

#!/bin/bash

Echo "dissenyant el firewall..."

Esborrem les regles que poden existir

Iptables -F

Iptables -X

Iptables -Z

...

Seguretat en xarxa

- **IPTABLES**. *Exemple:*

...

Establim les polítiques per defecte

Iptables -P INPUT ACCEPT

Iptables -P OUTPUT ACCEPT

Iptables -P FORWARD DROP

Filtratge de paquets

Permetre accés des de la pròpia màquina

Iptables -A INPUT -i lo -j ACCEPT

Permetem accés als ports 80 (web) i 20 i 21 (ftp)

Iptables -A INPUT -i eth0 -p tcp -dport 80 -j ACCEPT

Iptables -A INPUT -i eth0 -p tcp -dport 20:21 -j ACCEPT

...

Seguretat en xarxa

- **IPTABLES**. *Exemple:*

...

Permetem accés al nostre amic a la base de dades (port 3306)

```
Iptables -A INPUT -i eth0 -p tcp -s X.X.X.X -dport 3306 -j ACCEPT
```

Tanquem els ports de gestió (sota el 1024)

```
Iptables -A INPUT -i eth0 -p tcp -dport 1:1024 -j DROP
```

```
Iptables -A INPUT -i eth0 -p udp -dport 1:1024 -j DROP
```

Tanquem el port de mysql perquè ningú més pugui accedir (el nostre amic mai arribarà a aquesta regla fent una sol·licitud al mateix port)

```
Iptables -A INPUT -i eth0 -p tcp -dport 3306 -j DROP
```

Seguretat en xarxa

- **IPTABLES**. *Exemple:*
- *Per testejar si el nostre script aplica les regles correctament:*
 - *iptables -nL*

Seguretat en xarxa

- **IPTABLES**

Per guardar les nostres regles de firewall al sistema:

Scripts **iptables-save/iptables-restore**

- » `iptables-save > /root/iptables-save.out`
- » `iptables-restore -c /root/iptables-save.out`

Pràctica 7 - Iptables

- **IPTABLES**
 - *tcpdump* per visualitzar el tràfic.
 - *iptables* per:
 - filtrar paquets a la interfície local.
 - Bloquejar ports.
 - Redireccionar ports.