

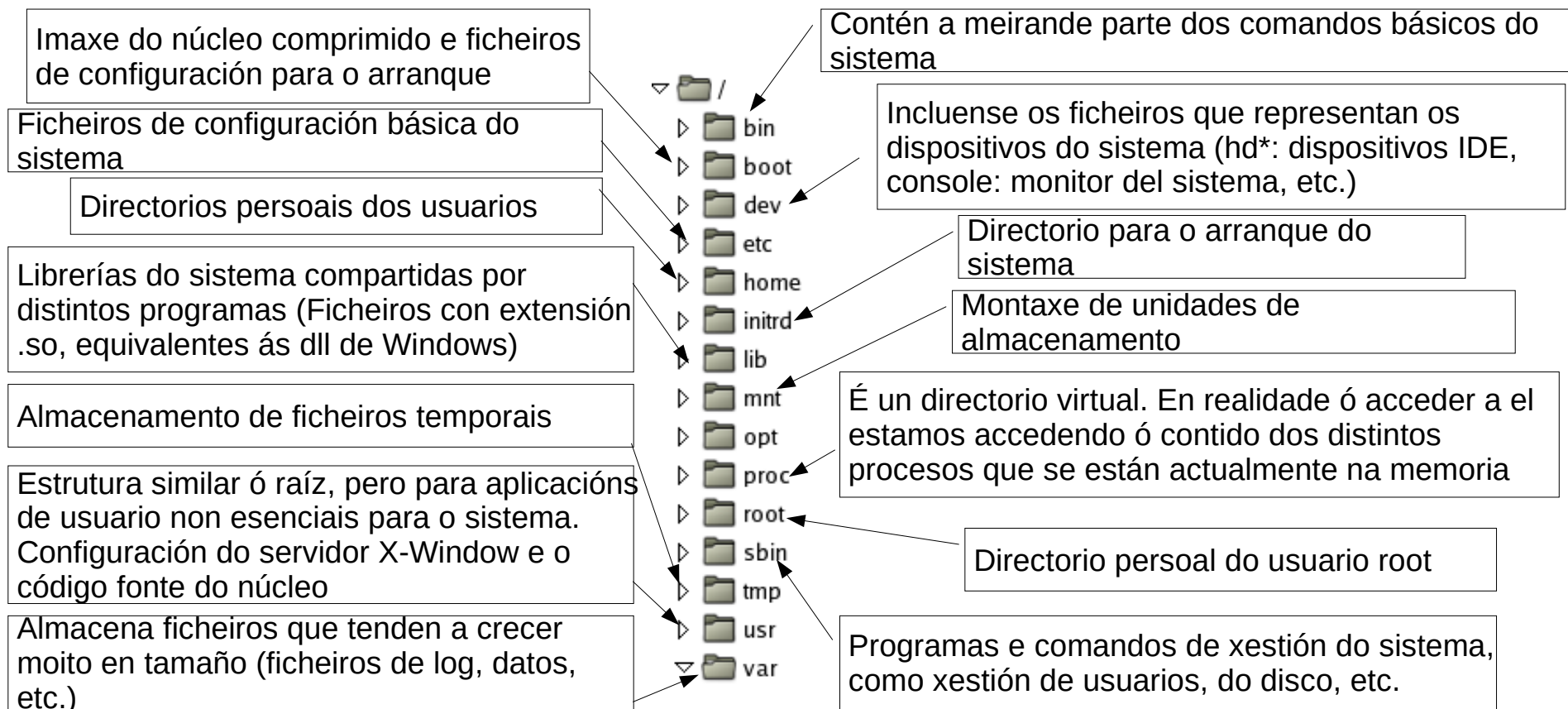
Administración básica de Linux

Copyright (c) 2005 Antonio de Andrés Lema, Vanesa Resúa Eiras e Carlos Carrión Álvarez.
Garántese permiso de copiar, distribuir e/ou modificar este documento baixo os termos da “GNU Free Documentation License”, versión 1.2 ou calquera versión publicada máis tarde pola Free Software Foundation; sen seccións invariantes, textos de portada, nen de contra portada. Unha copia de esta licencia se pode atopar en <http://www.gnu.org/licenses/fdl.txt>
Rógase notificar melloras, suxerencias e comentarios sobre este documento a aalema@edu.xunta.es

Manexando ficheiros e directorios



- O sistema de ficheiros de Linux consiste nunha serie de directorios que poden conter ficheiros ou outros directorios. A árbore parte dun directorio especial chamado directorio raíz e representado co carácter /.
- Linux contén unha estrutura predefinida de directorios no sistema:





¿Onde estou?: pwd

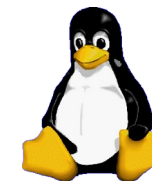
- Cando estamos utilizando o intérprete de comandos sempre temos un directorio de traballo, o directorio no que nos atopamos nese momento.
- Para saber cal é o directorio de traballo, utilizamos o comando pwd

The screenshot shows a terminal window titled 'antonio@Linux1.antonio.es: /home/antonio'. The menu bar includes 'Ficheiro', 'Editar', 'Ver', 'Terminal', 'Ir', and 'Axuda'. The command prompt is '[antonio@Linux1 antonio]\$'. The user has entered 'pwd', and the output is '/home/antonio'. A text box with a black border contains the text 'Directorio de traballo; ó inicio é o directorio home do usuario', with an arrow pointing to the output of the 'pwd' command.

```
antonio@Linux1.antonio.es: /home/antonio
Ficheiro  Editar  Ver  Terminal  Ir  Axuda
[antonio@Linux1 antonio]$ pwd
/home/antonio
[antonio@Linux1 antonio]$
```

Directorio de traballo; ó inicio é o directorio home do usuario

- Igual que noutros sistemas operativos, cando fagamos referencia a elementos do sistema de ficheiros (xa sexan ficheiros ou directorios) poderemos facelo de dúas formas:
 - Relativa: Indicamos unha ruta a partir do directorio de traballo.
 - Absoluta: Indicamos unha ruta completa ó elemento a partir do directorio raíz. Comezan por /



¿Que hai no directorio?: ls

- O comando ls permítenos visualizar os directorios e ficheiros que hai dentro dun directorio (por defecto no directorio de traballo).

```
antonio@Linux1.antonio.es: /home/antonio
Eicheiro  Editar  Ver  Terminal  Ir  Axuda
[antonio@Linux1 antonio]$ ls
Desktop/  Documents/  evolution/  nuevo/  tmp/
[antonio@Linux1 antonio]$ ls Desktop/
Home.desktop  Trash/  Welcome.desktop
[antonio@Linux1 antonio]$ ls /home/antonio/Desktop/
Home.desktop  Trash/  Welcome.desktop
[antonio@Linux1 antonio]$
```

No directorio actual hai 5 subdirectorios

Vemos o contido do subdirectorio Desktop indicando a ruta relativa

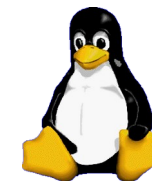
Vemos o contido do subdirectorio Desktop indicando a ruta absoluta

- O intérprete de comandos permite introducir metacaracteres nas rutas que se lles pasen ós comandos:
 - *: Cero ou varios caracteres.
 - ?: Un caracter.
 - [...]: Algún dos caracteres indicados (Por exemplo [ahc])



¿Que hai no directorio?: ls

- Os intérpretes de comandos permiten pasarlle parámetros ós comandos:
 - Os parámetros introdúcense despois do comando co símbolo –
 - Pódense combinar distintos parámetros.
 - Non importa a orde na que se introduzan os distintos parámetros do comando.
- O comando ls acepta moitos parámetros, os máis destacables son:
 - l: Mostra un listado completo con características dos ficheiros e directorios
 - i: Mostra os números de i-nodo dos ficheiros e directorios
 - a: Mostra tamén os ficheiros e directorios ocultos (empezan por .)
 - h: Mostra os tamaños en formato humano.
 - R: Listado recursivo.



¿Que hai no directorio?: ls

- Invocamos o comando ls con parámetros:

Tamaño total
dos ficheiros e
directorios
listados

```
[antonio@Linux1 antonio]$ ls -liha
total 168K
339167 drwxr-xr-x 26 antonio antonio 4,0K Set 10 11:46 ./
239521 drwxr-xr-x 3 root root 4,0K Set 6 20:21 ../
339282 -rw----- 1 antonio antonio 902 Set 10 10:32 .bash_history
339200 -rw-r--r-- 1 antonio antonio 24 Set 6 20:21 .bash_logout
339201 -rw-r--r-- 1 antonio antonio 191 Set 6 20:21 .bash_profile
339202 -rw-r--r-- 1 antonio antonio 124 Set 6 20:21 .bashrc
339231 -rw-rw-r-- 1 antonio antonio 14 Set 6 20:22 .desktop
275391 drwxrwxr-x 3 antonio antonio 4,0K Set 7 10:20 Desktop/
339221 drwxr-xr-x 2 antonio antonio 4,0K Set 6 20:22 Documents/
339212 -rw-rw-r-- 1 antonio antonio 0 Set 6 20:22 .drakfw
339284 -rw----- 1 antonio antonio 16 Set 6 20:24 .esd_auth
467010 drwx----- 6 antonio antonio 4,0K Set 6 20:37 evolution/
339214 -rw-rw-r-- 1 antonio antonio 400 Set 6 20:22 .fonts.cache-1
403074 drwxr-xr-x 5 antonio antonio 4,0K Set 7 13:45 .galeon/
339220 drwx----- 7 antonio antonio 4,0K Set 10 11:42 .gconf/
339213 drwx----- 3 antonio antonio 4,0K Set 10 11:46 .gconfd/
211517 drwx----- 2 antonio antonio 4,0K Set 7 10:09 .gftp/
402276 drwx----- 5 antonio antonio 4,0K Set 7 05:14 .gnome/
```

Número
de i-nodo

Tipo de
ficheiro e
permisos

Número de enlaces
sobre o ficheiro ou
directorio

Usuario
propietario

Grupo
propietario

Tamaño

Data de última
modificación

Nome

Tipos de ficheiros: Creando ficheiros e directorios



- En Linux existen catro tipos de ficheiros:
 - Ficheiros normais (-):
 - Son ficheiros con datos, xa sexan de texto ou binarios. Teñen un nome e poden ter unha extensión, aínda que Linux non asocia tipos de ficheiros nin aplicacións segundo as extensións dos mesmos.
 - Os ficheiros de datos xéranse dende distintas aplicacións (editores de texto, compiladores, etc.). Podemos crear un ficheiro baleiro co comando touch
 - Directorios (d):
 - Os directorios en Linux son ficheiros dun tipo especial, que como información conteñen unha lista de entradas cos números de i-nodo e nome dos ficheiros e directorios que conteñen (táboa de entradas de directorio).
 - Para crear un directorio, usaremos mkdir nome_directorio.



¿Como me movo?: cd

- O comando cd permítenos cambiar o directorio de traballo. Simplemente o invocaremos como cd rutaDirectorio para movernos a ese directorio.
- Hai unha serie de valores especiais que podemos indicar na ruta do directorio:
 - cd . , cd..: Todo directorio en Linux ten a entrada . que apunta ó propio directorio e a entrada .. que apunta ó directorio superior.
 - cd : Se non indicamos ningunha ruta levaranos ó noso directorio home.
 - cd -: Permite volver atrás; ó último directorio de traballo.

```
[antonio@Linux1 antonio]$ pwd
/home/antonio
[antonio@Linux1 antonio]$ cd ..
[antonio@Linux1 home]$ pwd
/home
[antonio@Linux1 home]$ cd /bin
[antonio@Linux1 bin]$ pwd
/bin
[antonio@Linux1 bin]$ █
```

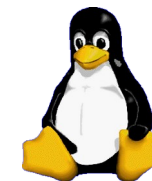



Copiando e movendo

- Podemos copiar ficheiros dunha ruta a outra co comando `cp`. Usaremos `cp` orixe destino.
- O parámetro `-R` permítenos copiar recursivamente directorios completos.
- Se en lugar de copiar, queremos mover ficheiros ou directorios, usaremos `mv` orixe destino.

Visualizar ficheiros

- Desde o intérprete de comandos podemos visualizar o contido dos ficheiros con dous comandos:
 - `more`: Mostra o contido do ficheiro de forma paxinada.
 - `cat`: Permite concatenar ficheiros e mostralos pola saída estándar (a consola). Se só lle pasamos un ficheiro, mostra por pantalla o contido do mesmo.



Borrando ...

- Para borrar ficheiros utilizamos o comando rm. Os parámetros máis destacables deste comando son:
 - R ou -r: Permite borrar directorios, borrando previamente todo o contido dos mesmos de xeito recursivo.
 - i: Pide confirmación ó usuario antes de cada borrado.
 - f: Non pide confirmación nin amosa erros no proceso.
- Se queremos borrar directorios baleiros podemos utilizar o comando rmdir.

```
[antonio@Linux1 antonio]$ ls
Desktop/      Documents/      EnlDuroExemplo FichExemplo  tmp/
DirExemplo/  EnlDebilExemplo@ evolution/      nuevo/
[antonio@Linux1 antonio]$ mv FichExemplo DirExemplo/
[antonio@Linux1 antonio]$ ls DirExemplo/
FichExemplo
[antonio@Linux1 antonio]$ rm -r DirExemplo/
rm: ¿descender al directorio "DirExemplo/"? (s/n) s
rm: ¿borrar el fichero normal baleiro "DirExemplo//FichExemplo"? (s/n) s
rm: ¿borrar el directorio "DirExemplo/"? (s/n) s
[antonio@Linux1 antonio]$
```

Xestionando o acceso: Os permisos



- Linux permite configurar os privilexios e restriccións que os usuarios teñen ó acceder ó sistema. Esta configuración baséase nunha serie de permisos sobre os ficheiros que se outorgarán ós usuarios e grupos.
- Todo ficheiro ten un usuario propietario (por defecto, o usuario que o creou), e un grupo propietario (por defecto, o grupo principal do usuario que o creou).
- Os permisos dos ficheiros defínense a tres niveis:
 - Permisos para o usuario propietario.
 - Permisos para o grupo propietario.
 - Permisos para todos os usuarios.
- Hai que ter en conta que sempre se aplicarán os permisos máis concretos; é dicir; se o usuario coincide, os de usuario, se non os de grupo, e se non os de todos, aínda que os de usuario fosen máis restrictivos.

Xestionando o acceso: Os permisos



- Para cada un dos niveis defínense tres tipos de permisos:
 - Permiso de lectura (r): Para un ficheiro, permite a lectura dos seus datos. Para un directorio, permite listar o seu contido.
 - Permiso de escritura (w): Para un ficheiro, permite escribir ou modificar datos nel. Para un directorio permite cambiar o listado do directorio, é dicir, engadir ou eliminar nel ficheiros e directorios.
 - Permiso de execución (x): Para un ficheiro, permite executalo. Para un directorio, permite facelo activo (facer cd a ese directorio). Se se restrinxe este permiso, propágase ós subdirectorios.

```
[antonio@Linux1 antonio]$ ls -l
total 20
drwxrwxr-x 3 antonio antonio 4096 sep  7 10:20 Desktop/
```

Usuario propietario

Grupo propietario

Permisos do ficheiro:

- Usuario propietario:rwX (Pode ler, escribir e executar)
- Grupo propietario:rwX (Pode ler, escribir e executar)
- Todos os usuarios:r-x (Pode ler e executar, pero non escribir)

Xestionando o acceso: Os permisos



- Polo tanto para interpretar os permisos dun ficheiro, deberemos ler a cadea de permisos de tres en tres para obter os permisos do propietario, grupo propietario e resto dos usuarios, seguindo este orde.
- Podemos cambiar os permisos dun ficheiro con **chmod**:
 - Só pode cambiar os permisos dun ficheiro o usuario propietario ou o root.
 - Teclearemos chmod novosPermisos ficheiro, onde os novos permisos poden ser indicados de dúas formas:
 - Mediante unha máscara: Os novos permisos escríbense con tres números en octal, de forma que os permisos de usuario, grupo e outros suman 4,2 e 1 respectivamente:

<u>USUARIO</u>			<u>GRUPO</u>			<u>OUTROS</u>		
r	w	x	r	w	x	r	w	x
4	2	1	4	2	1	4	2	1

Así **chmod 725 fich** establece os permisos: rwx-w-r-x

Xestionando o acceso: Os permisos



- Notación simbólica: Nesta notación indícanse os permisos que se quitan ou se engaden na forma ugoa+ -=rwx, onde:
 - ugo indica se se modifican os permisos de usuario, grupo ou outros
 - a indica todos os usuarios (equivale a ugo)
 - +- se se engaden ou quitan permisos
 - rwx indican os permisos a modificar
 - = asigna os permisos

`chmod ug+r, o-wr fich`: engade ó usuario é ó grupo permisos de lectura, e o resto quítalles os permisos de lectura e escritura.

`chmod u=rwx, go=r fich`: asigna ó usuario todos os permisos é ó grupo e outros permisos de lectura.



A redirección da e/s

- En Linux todos os programas abren cando arrancan tres fluxos para a entrada/saída: Un para a entrada (stdin-0), outro para a saída (stdout-1) e outro para as mensaxes de erro (stderr-2).
- Por defecto, estes fluxos están enlazados co terminal do usuario, tomando a entrada do teclado e mostrando a saída e erros pola pantalla.
- O intérprete de comandos permite redirixir estes fluxos, de forma que en lugar de enlazar un terminal enlacen cun ficheiro.
 - comando > ficheiro: Redirección da saída do comando a un ficheiro. Se o ficheiro xa existe sobrescríbese, se non, crease.
 - comando >> ficheiro: Redirección da saída do comando ó ficheiro. Se o ficheiro xa existe, engádese á saída do comando, se non, crease.
 - comando < ficheiro: Redirección da entrada do comando a un ficheiro.

A redirección da e/s



- Tamén podemos indicar o fluxo a redirixir utilizando o seu número (0,1,2), por exemplo:

`ls -li 2>error.log`

Rediriximos o erro estándar
ao ficheiro "error.log"

- Podemos redirixir un fluxo a outro usando o carácter &:

`ls -li 1>&2`

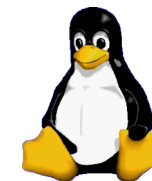
Rediriximos a saída estándar ao erro estándar

`ls -li 2>&1`

Rediriximos o erro estándar á saída estándar

- Por último, podemos redirixir calquera fluxo o dispositivo `/dev/null` se queremos ignorar completamente ese fluxo.

`ls -li 2>/dev/null`



A redirección da e/s

- O intérprete de comandos tamén permite enlazar a saída dun comando coa entrada doutro comando con comando1 | comando2 (pipe, tubería, cano).
- Algúns exemplos:

```
[antonio@Linux1 antonio]$ ls > listado.txt  
[antonio@Linux1 antonio]$ sort < listado.txt
```

Rediriximos a saída do comando ls; o listado resultante almacénase no ficheiro listado.txt

O comando sort toma un texto de entrada por teclado e sácao pola pantalla ordenado. Rediriximos a entrada para que ordene o ficheiro listado.txt

```
[antonio@Linux1 antonio]$ sort < listado.txt > listadoOrdenado.txt  
[antonio@Linux1 antonio]$ ls / >> listado.txt  
[antonio@Linux1 antonio]$ echo "Xenero un ficheiro cunha mensaxe de texto" > nota.txt  
[antonio@Linux1 antonio]$ ls | more
```

Agregamos no ficheiro listado.txt o listado do directorio raíz.

```
Desktop/  
Documents/  
evolution/  
listado.txt  
tmp/  
[antonio@Linux1 antonio]$
```

Usamos unha tubería para paxinar con more o listado xerado polo ls

O comando echo mostra unha mensaxe por pantalla. Rediriximos a súa saída para xerar un ficheiro de texto cunha mensaxe.

Rediriximos agora tamén a saída de sort para que o listado ordenado se garde nun ficheiro



Comandos útiles en tuberías

- Hai unha serie de comandos que podemos destacar polo seu uso frecuente enlazados con outros comandos, como son:
 - `sort`: Este comando ordena o texto recibido pola entrada estándar ou que le de un ou varios ficheiros e o mostra ordenando alfabeticamente as liñas na saída estándar
 - `wc`: Mostra o número de liñas (-l), palabras (-w) e caracteres (-m) ou bytes (-c) da entrada ou de ficheiros
 - `cut`: Este comando corta parte dun texto de entrada e o mostra na saída estándar. Pode cortar uns determinados bytes (-b), caracteres (-c) ou campos do texto (-f, e indicaremos o delimitador dos campos con -d). Indicaremos que elementos queremos mostrar de cada liña do texto de entrada a continuación:
 - `N`: Mostra só n-ésimo o elemento
 - `N-`: Dende o n-ésimo ao final da liña
 - `N-M`: Dende o n-ésimo ao m-ésimo
 - `-M`: Dende o comezo da liña ao m-ésimo

Exemplos:

```
cut -c 5- proba.txt
```

```
cut -f 2 -d " " proba.txt
```



Comandos útiles en tuberías

- grep: Este completo comando permite buscar patrones dentro dun texto da entrada estándar ou de ficheiros. Por defecto grep mostra na saída as liñas atopadas que conteñen o patrón introducido, precedidas do nome do ficheiro no que se atopan se procede:

grep linux *

Busca nos ficheiros do directorio actual liñas que conteñen a palabra "linux"

- Este comportamento pódese modificar cos seguintes parámetros:
 - -c: Mostra o número de liñas que conteñen o patrón
 - -i: Ignora maiúsculas e minúsculas
 - -n: Mostra o número de cada liña dentro do seu ficheiro
 - -f fich: Toma o patrón dun ficheiro
 - -v: Mostra as liñas que non teñen o patrón
- Algunhas expresións regulares que se poden usar con grep son:
 - []: O caracter pode ser un do conxunto indicado: [abc] [a-c]
 - O caracteres . e ? concordan con calquera caracter, e * con varios