

Entorno de script
o funciones

Shell (#2, ##, ##)

#? → Sobre la ejecución anterior (0-bien).

→ Número de argumentos pasados al script o comando.

##* → Contiene la lista de argumentos. "#*" → Todos los parámetros forman una cadena.

Bash ('', "", \) → Para interpretar caracteres especiales y variables
permite anidación.

' → texto literal

" → deja la expansión de \$, \, `` → Es lo mismo que '\$1'.

'' → comillas graves para la expansión de comandos

\ → trata el siguiente carácter literal.

Los comandos se buscan en el \$PATH variable (PATH = "\$PATH:\$") donde el directorio actual.

Metacaracteres/comodines → Se aplica a nombres de ficheros y directorios (ls, echo, cat, tail, head).

* → coincide con cualquier secuencia de caracteres que no empieza por . (incluida cadena vacía).

? → Representa cualquier carácter (un carácter).

[] → coincide con cualquier carácter dentro de un conjunto o rango ([]) → excluye [a-z] → incluye [a-zA-Z]

Expresiones regulares → Se aplica al contenido del fichero (por líneas) (grep, sed, ...).

^ → Principio de línea.

[] → coincide con cualquier carácter dentro de un conjunto o rango.

[^] → Excluye lo que hay en su interior.

* → Cero o más repeticiones del carácter o grupo anterior.

. → Coincide con cualquier carácter excepto salto de línea.

\$ → Final de línea.

\| \) → Expresión marcada. Agrupa una expresión regular para luego referenciarlo. (\1...)

grep → Busca texto en un archivo(s) varios). grep [opc.] [expresión regular] fichero(s)

-v → Excluye las líneas que coinciden.

sed → Stream Editor (no modifica el fichero). sed [opc.] 'acción/expresión/acción' fichero(s)

-e → permite especificar el comando.

-f → lee comando desde un fichero.

solo se usa el primer archivo como fichero de comando.

! uso de (!) antes de acción hace lo contrario.

'acción/expresión/acción' fichero(s)

se sustituye → elimina

añade → imprime

's/exp./reemplazo/g'

g → aplica a todos los coincidencias

i → inserta el contenido de un archivo después de los líneas que coinciden con el patrón

awk → Filtro para ficheros de texto programable awk, [opc.] parte donde donde [fichero] fichero
Cuenta desde Selecciona campos y registros (líneas) de un fichero de entrada y el resultado se muestra
el 1. en la salida estandar.

- #0 → Representa la linea completa.
- #1 → Primer campo.
- #30 → Tercer campo.

NR → Número del registro actual.

NF → Número de campos del registro.

FILENAME → Nombre del fichero de entrada.

Si no se especifica [acción], awk ejecuta print.

Si se omite la parte donde la acción se aplica a todos los registros.

Parte donde

dir [acción] → se aplica a dir

En la parte donde para usar expresiones regulares deben ir entre / /.

dir1, dir2 [acción] → se aplica al rango dir1-dir2.

BEGIN y END permite realizar acciones antes o despues de procesar las líneas.

awk 'BEGIN[acción] [acción] [END[acción]]' fich

[opc.]

-f → file, como 'parte-donde [acción]'.

-F 'x' → Field, cambia el delimitador de campo a 'x'.

[acción]

de escritura

print

printf → como en C

de asignación de control de flujo

{ ; ; }

{ while() ; ; }

{ for(; ;) ; }

funciones

length(s) → Devuelve la longitud de la cadena de caracteres.

substr(s,p) → Extrae una subcadena de s comenzando en la posición p, hasta el final.

substr(s,p,n) → Extrae una subcadena de s, comenzando en la posición p, de n caracteres. * Los indices empiezan en 1.

index(s, st) → Devuelve la posición inicial de la primera aparición de st en s.
Si no se encuentra devuelve 0.

n = split(s,v,sep) → Divide la cadena s en partes usando sep como delimitador y las almacena en el array v, devuelve el numero de partes resultantes.

Caracteres individuales o rangos.

tr tr [opc.] Alcaracteres1 Alcaracteres2

- d → Elimina los caracteres indicados.
- c → Selecciona los caracteres que no están en la lista 1.
- s → Reduce los caracteres repetidos consecutivos especificados en lista 1 a un solo carácter.

cut → Selecciona columnas o campos específicos de un archivo o de la entrada estandar.

- d → Especifica el delimitador (por defecto tabulador).

-f → Selecciona los campos basados en el delimitador (field).

- e → Selecciona caracteres específicos (x, y, t) o rangos ($x-f$). $x-f$ → hasta el final.

tac → Muestra las líneas del archivo en orden inverso.

rev → Invierte el orden de los caracteres de cada linea.

sort → Ordena las líneas de un archivo (según ASCII).

- r → Ordena en orden inverso.

- n → Ordena numéricamente.

- u → Ordena las líneas eliminando las que son iguales.

uniq → Elimina las líneas duplicadas consecutivas.

- c → Número de veces que aparece cada linea.

- d → Muestra solo las líneas duplicadas.

paste → Junta archivos linea por linea.

- d → Especifica un delimitador diferente al tabulador.

comm → Compara dos archivos ordenados linea por linea

- 1 -2 -3 → Omite la columna(s) seleccionada(s).

Líneas exclusivas del 1er fichero	Líneas exclusivas del 2º fichero	Líneas comunes en ambos
-----------------------------------	----------------------------------	-------------------------

join → Combina dos archivos linea por linea según un campo común → ordenados según

- 1 -2 → Especifica que campo utiliza de cada archivo para hacer la combinación.

- t → Especifica el delimitador que separa los campos.

- o → Permite seleccionar de qué archivo se muestra que columna.

Comando test

- test permite evaluar expresiones:
 - En los que se comparan números o cadena de caracteres (strings)
 - En los que se comprueban atributos de ficheros/atributos.
- test siempre devuelve un valor booleano.
 - 0 true.
 - Distinto 0 false.

test expresión ⌈ [expresión] ⌋
Espacios
no funcionan.

test	
Números enteros	strings
x -eq y x equals y	x = y x igual a y
"-ge" "greater equal y"	x != y x distinto de y
"-gt" "greater than y"	x no es null
x -le y x less equal y	-n x longitud de x es mayor que 0.
x -lt y x less than y	-z x la longitud de x es igual que 0.
x -ne y x non equal y	

Ficheros		
-d nom	directory	
-f nom	file	
permisos	{-r nomh -w nomh -x nomh -s nomh}	read write execute longitud distinta de 0

Operadores lógicos	
! expr	expr no es verdadera
e1 -a e2	e1 y e2 son verdaderas
e1 -o e2	e1 ó e2 son verdaderas

Recetas

Sesión 9 AWK → Empieza por el 1 (la diferencia de C)

- sentencias del programa orden:

de escritura

print

printf

de asignación

de control de flujo

if else

while

for (; ;)

for (in

• Funciones

-length(s) → Devuelve la longitud de la cadena de caracteres(string)s.

-substr(s, p) → Extrae una subcadena de s, comenzando en la posición p hasta el final.

-substr(s, p, n) → Similar a substr(s, p) pero extrae exactamente, ^{Los índices comienzan} en 1.
n caracteres desde p. o si el numero de caracteres

-index(s, s1) → Devuelve la posición (índice) de la primera ^{excede la longitud de la cadena, devuelve solo las} caracteres disponibles.

en s. → Si s1 no se encuentra en s, ^{aparición de la cadena s1} Devuelve 0.

-n = split(s, v, sep) → Divide la cadena s en un array v, utilizando el separador sep.

Devuelve el ~~el~~ número de elementos en el array.

Si no se especifica sep, utiliza espacios (tabulaciones).

split devuelve el número de elementos

Redirección de entrada

↳ ^{centinela} → Último elemento.

tr acción AEIOU ↳ YA

palabra centinela

tee

Saca dos copias de los caracteres que lee. Una copia va a la salida estandar, y la otra al fichero.

Siempre y cuando no este redirigida.

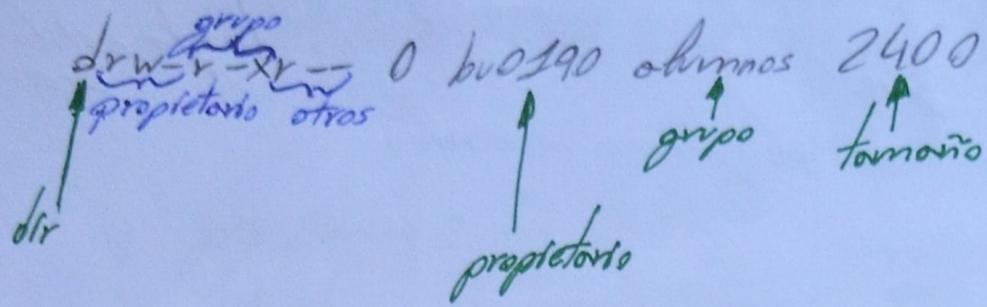
Agrupación de comandos

• Los parentesis () permiten agrupar comandos.

• El carácter ; permite separar comandos.

Permisos

Permisos y propiedades de ficheros y directorios:
ls -l



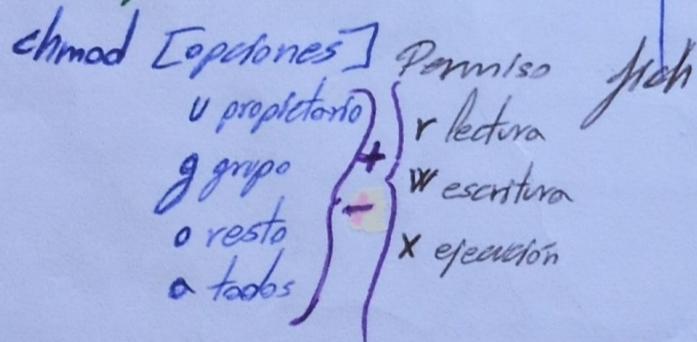
Permisos en ficheros

- r: permite leer
- w: permite modificar
- x: permite ser ejecutado

Permisos en ~~archivos~~ directorios

- r: permite listar(ls)
- w: permite borrar o añadir nuevos ficheros o directorio.
- x: permite ser atravesado(cd)

Cambio de permisos



 r:4
 w:2
 x:1

 chmod 740
 proprietario rwx
 |
 +----- grupo r
 |
 +----- otros

Umask

Permite establecer los permisos al crear los ficheros y directorios.
- máscara para root (el que gestiona la máquina).

0022
máscara de permisos a no activar.

Fichero por defecto 666 (rw- rw- rw-)

Directorio por defecto 777 (rwx rwx rwx)

Session 10

Procesos

• programa en ejecución
conjunto de instrucciones

`ps -process`, `status` (como `cptkra`) de lo que hay en ejecución
`pstree` -arbol de procesos con sus hijos. -efecto Estilo Sistémico

~~postree~~ se ve como son los procesos de forma dinámica

pid → process identifier.

FTy o terminal asociado al proceso.

TIME → tiempo de CPU consumido por el proceso.

TIME → tiempo de CPU consumido por el proceso.
CMD → nombre del comando que crea el proceso.

KILL → Acaba con la ejecución de un proceso

Acaba con la ejecución de un proceso interactivo

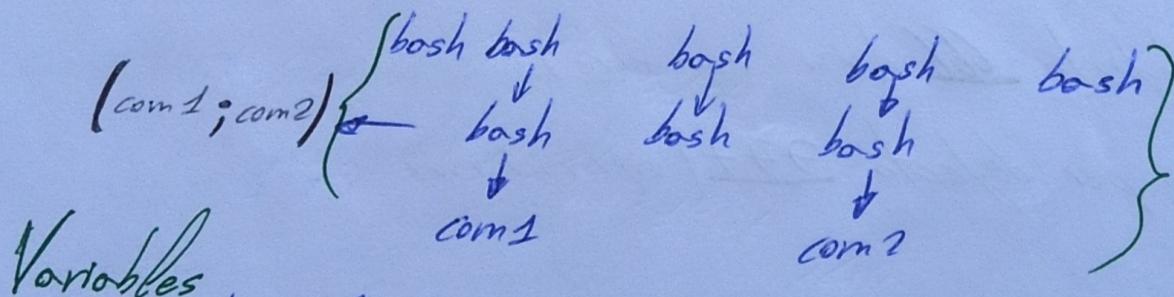
Kill [options]

KILL -9 ← El programa termina si o si.

Creación de procesos

Foreground  & { [K] 21465
nº de proceso identificador de tarea
Lo pone en segundo plano

Tuberías (1) o procesos concurrentes con el mismo padre.



Se puede utilizar `$` para delimitar el nombre de una variable.

Variables de entorno

VARIABLES DE ENTORNO
Los comandos están en el PATH y se busca el comando donde dice PATH.
Las variables no predefinidas no se heredan en los bash hijos.

Con export se pueden exportar variables a un bash hijo

• Lo que se exporta es una copia, si se modifica, no afecta al original (de otro).

Read

Compresión de ficheros

Compresión de ficheros:

compress [opciones] fich
↑
extensión .z

gzip [opciones] fich
↑
extensión .gz -9 → máxima compresión

Descompresión

uncompress

gunzip

Ver ficheros comprimidos (z y gz)

zcat fich

zless fich

o more

Estructuradois arbol ficheros.

tar → creación de un fichero con una estructura conteniendo un arbol de directorios/ficheros

tar [opc.] fich.tar [origen]

-c
-t
-x
-v
-z

} fich

Sesión 11

Caracteres parámetros de un script.

\$0 → nombre del script

\$1 → primer parámetro del script

\$2

:

\$n → n-ésimo parámetro del script

→ nº de parámetros del script

** → lista de los parámetros del script.

Case

ose
P ser de operaciones si se cumple una condicón de tipo cadena de caracteres.

case expression In

option 1)

sent 1

sent n 33

Epopón 2)

sent 1

sentences]

— opción si no es todo lo anterior.

sent L

1

Sent in

esac no hace falta.

esac no more gotta.

If for while } No other has ;

Mirar medida $\#() = 11$
 Variables y parámetros } Sesión 11
 $[v_{uu}] = \text{test}$ } página 9

Caracteres y parámetros de un Script

$\* \neq " $\* "
listo de
parámetros & único elemento

Fundones en un Script

nombref() §3 & function nombref() §3 - Definitions

~~s/v~~ parámetros \rightarrow nombrefj \rightarrow nombrefx part ... part } Invocación

Caracteres especiales BASH

" Todo lo que encuentra dentro de las comillas simples se interpreta literalmente, sin ninguna expansión o interpretación especial.

"" Las comillas dobles permiten la expansión de variables y protege los espacios. La bash solo interpreta #, \, int permite la expansión de comandos (como date). acentos graves (no confundir con comillas simples).

\ Backslash se utiliza para "escapar" un carácter especial, haciendo que se interprete textualmente.

Comandos (con ; se pueden realizar varios comandos en la misma línea).

ls Lista archivos y directorios.

ls -l Muestra detalles (permisos, propietario, tamaño, fecha,

ls -a Incluye archivos ocultos (que comienzan por .).

ls -R Lista el contenido de forma recursiva.

wc Cuenta líneas, palabras y caracteres.

wc -l Cuenta solo las líneas.

wc -w Cuenta solo las palabras.

wc -c Cuenta solo los caracteres.

cat Muestra o concatena archivos.

more Muestra el contenido de archivos página por página.

espacio Pasa a la siguiente página.

enter Avanza una linea.

less Less is more.

espacio

enter

arrows Navegar hacia arriba y abajo.

q salir.

head Muestra las primeras líneas de un archivo (10 por defecto).

head -n Muestra las primeras n líneas.

tail Muestra las últimas líneas de un archivo (10 por defecto).

tail -n Muestra las últimas n líneas.

pwd Print work directory

echo Imprime texto en la terminal.

grep Busca texto en un archivo.

grep -v Excluye los líneas que coinciden.

nl Number lines

Metacaracteres / Comodines — Se optan a nombres de ficheros y directorios (ls, echo, tail, head, cat).

* Representa cualquier secuencia de caracteres que no empleen por . puede representar la secuencia vacía.

? Representa enquier corocter (un corocter).

[] Coincide con cualquier carácter dentro de un conjunto, o rango.
[aeiou] [a-z]

Manipulación de ficheros

cp Copiar archivos y directorios

cp -r copia directorios recursivamente.

ln Crea enlaces (hard links o symbolic links) → se deben eliminar todos para borrar su info.

In -s Crea un enlace simbólico. ← Es un "puntero" al archivo original; si el original se borra, el enlace simbólico se rompe.

mv Mover o renombrar archivos y directorios. enlace simbólico se rompe.

rm Eliminar archivos o directorios.

`rm -r` Eliminar directorios recursivamente.

rm - f Eliminar de forma forzosa.

rm -i Pregunta antes de eliminar cada archivo.

Directorio (mkdir)

- Directorio actual
 - Directorio padre

Ruta absoluta ← Ruta completa desde el directorio raíz (II)

Ruta rectiva → Depende del lugar en el que estés ubicado, no empieza por 1.

cd Change directory

Disk usage

✓ -h Muestra el formato de manera legible (MB, GB, etc.).

dv -s Muestra el tamaño total del directorio especificado.

Muestra el uso de espacio para todos los archivos y directorios.

Editor Vi

Modo Inserción $\alpha, A, i, I, o, O, r, R, s$.

Modo comando esc. → x borra el carácter debajo del cursor del borra la palabra del cursor

:? Solr. :?! Solr forzosamente.

:w Guardar. :wg Guardar y saldr.

~~de~~ bono la poldra del curro

dd borra la linea del cursor

• (mala) deshaze el último combate.

W mueve el cursor a la siguiente palabra.

Expresiones regulares → Se opera al contenido del fichero (por líneas) (grep, sed,...).

^ El patrón está al principio de línea.

* Cero o más repeticiones del carácter o grupo anterior

\$ Busca el patrón al final de línea.

· Coincide con cualquier carácter excepto salto de línea.

[] Coincide con un carácter o al rango de dentro de los corchetes.

[^] Coincide con cualquier carácter excepto las indicadas dentro de los corchetes.

\ Se usa para escapar metacaracteres para que se interpreten de forma literal.

\` Expresión marcada. Agrega una expresión regular para luego hacer referencia a ese grupo.

Redirección

> Redirección de la salida estandar, creando o sobreescritbiendo.

>> Redirección de la salida, agregando.

< Redirección de entrada estandar, toma la entrada de un archivo.

<< Redirección de entrada, toma la entrada de la linea de comandos.

• Tras << se encuentra con el delimitador que se introducirá, tras eso leerá la entrada del terminal hasta encontrar dicho delimitador.

Tuberías (1).

Las tuberías (1) redirigen la salida estandar de un comando para que se convierta en la entrada estandar de otro comando.

Filtros.

cot Selecciona columnas o campos específicos de un archivo o de la entrada estandar.

cat -d Especifica el delimitador (por defecto tabulador).

cat -f Selecciona los campos basados en el delimitador (field).

cat -c Selecciona en rango de caracteres.

tail Muestra el contenido de un archivo al revés (líneas).

rev Invierte el orden de los caracteres de cada línea.

sort Ordena las líneas de un archivo. (Según ASCII).

sort -r Ordena en orden inverso.

sort -n Ordena numéricamente.

sort -u Ordena las líneas eliminando las que son iguales.

uniq Elimina líneas duplicadas consecutivas.

uniq -c Muestra en conteo de cuantas veces aparece cada linea.

uniq -d Muestra solo las líneas duplicadas.

paste Junta archivos linea por linea

paste -d Especifica un delimitador diferente al tabulador.

comm Compara dos archivos ordenados linea por linea.

comm -1 -2 -3 Muestra la columna 1, 2 o 3 dependiendo lo que se quiera comparar.

Líneas
exclusivas
del 1º
archivo

Líneas
exclusivas
del 2º
archivo

Líneas
comunes
entre
ambos
archivos

join Combina dos archivos linea por linea segun un campo comun. → Ordenados segun la Key.

join -1 -2 Especifica que campo utilizar de cada archivo para hacer la combinacion.

join -t Especifica el delimitador que separa los campos.

join -o Permite seleccionar de que archivo se muestra que columna.

pr Formatea un archivo para impresion

pr -l Especifica la longitud de pagina (número de líneas por pagina).

pr -h Añade un encabezado a cada pagina.

pr -t Quita el encabezado.

pr -n Numeros las líneas.

Scripts

Es un archivo que contiene comandos de la Bash, los comandos se procesan uno tras otro.

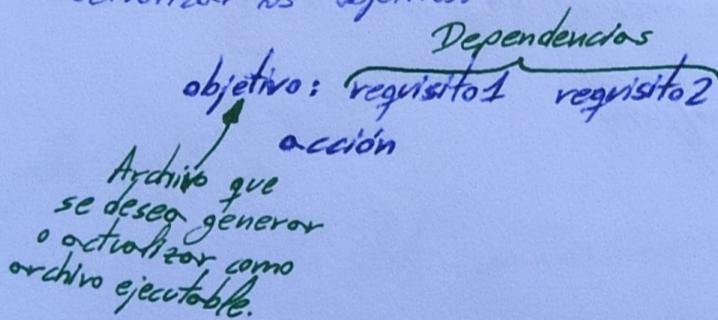
Suelen terminar en .sh se puede llamar a la bash `a.sh` para que lo interprete o hacer el archivo ejecutable chmod +x a.sh para ejecutarlo ./a.sh.

Make

Determina qué archivos necesitan ser recompilados o actualizados en un proyecto. Para hacer esto se basa en las fechas de modificación de los archivos.

Makefile

Es un archivo que contiene una serie de reglas que make sigue. Las reglas definen objetivos/archivos que se desean generar y sus dependencias junto con las acciones que se deben ejecutar para activarlos los objetivos.



Si la fecha del objetivo es anterior a la fecha más reciente de alguno de sus requisitos, se ejecuta la acción asociada para activar el objetivo.

sed & linea por linea (no modifica) opción para no la función

sed [opciones] 'comando' archivo

5 ||| 0 ||| "

-e
-o aplica todos los comandos del fichero a cada linea del fichero de entrada.

-d no selecciona la linea (no la muestra en pantalla).

& selecciona la linea (la repite)

slang2f sustituye arg1 por arg2 (solo una vez por linea).

s!arg1!arg2!g sustituye arg1 por arg2 (global),

& arg1 inserta el fichero arg1 después de la linea seleccionada

última linea \$.

El contenido entre slashes es una expresión regular

Si se pone m.! antes de la función (d,p,s,r) se efectúa al contrario

sed -e parte_dirección acción argumentos fichero

for

for var in lista ; misma linea

do sent1
sentn } sentencias en lineas
done

referenciación de lavariable
#variable

for var in lista ; do sent1 ; sentn ; done

Sed -f solo usa el primer archivo como fichero de parámetros, el segundo y consecutivos se concatenan por orden de entrada y se les aplican los parámetros.

Si no pones "programa" en awk, awk ejecuta \$print\$
(2 3)

dir \$accción\$ o Se aplica a dir
dir1, dir2 \$accción\$ o Se aplica al rango dir1 - dir2

\$accción\$ o se aplica a todos los registros

awk '\$print "=-", \$1, " -- ", \$3' dias
 ^ ^
 print blancos(,)

awk -f ficheros.awk regímenes
file

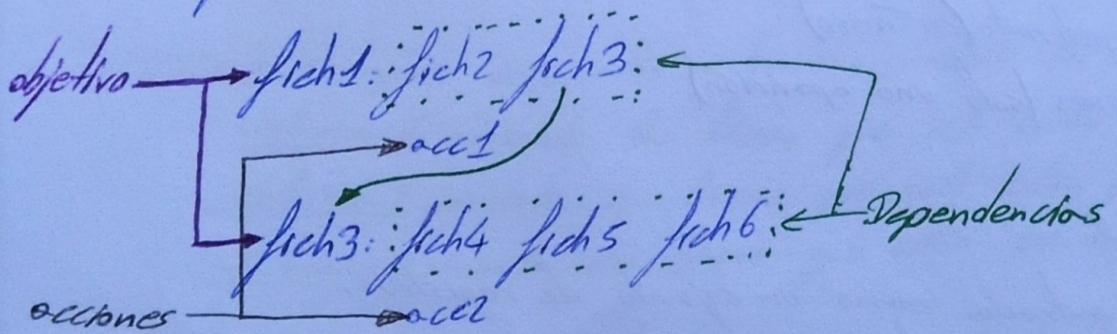
awk -Fx o cambia el delimitador de campos a x.
field

Comillas graves `` equivalente a \$() o \$() para meter en variables
la ejecución de algún comando.

Make.

make [opciones] [objetivo]

Makefile:



make
Equivalente
make fich1

make -n *
Señala las acciones que se ejecutarán (pero no las ejecutó).

awk → Filtro para ficheros de texto programable.
selecciona campos y registros/líneas) de un fichero de entrada y el resultado se muestra en la salida estandar.

awk -parametros parte donde programa fichero
"acción"

Sesión 7

Shell \$? ## \$*

\$? → Para ver si los scripts se ejecutan correctamente
Resultado de la ejecución 0 correcto (true)

número de parámetros del script

Otro valor entero para incorrecto (false)

\$* lista de parámetros del script

If expresión Booleana
then
sent1
:
sentn
else
fi

If expresión Booleana; then sent1; ... sentn; fi

Sed

sed -e parte-dirección acción parámetros
argumentos fichero

tr

tr [opciones] lista1caracteres1 [lista2caracteres2]
-d elimina los caracteres indicados
-c complemento del cambio indicado (contrario)
-s combina y quita repeticiones (solo una aparición)

Sesión 8

awk considera el fichero de entrada como un conjunto de registros.
-Por defecto, cada línea es un registro.

\$0 → campo 0
\$1
\$30

NR → Número del registro con el que se trabaja

NF → Número de campos del registro

FILENAME → Nombre del fichero de entrada

Parte donde

BEGIN
parte donde
END