

RESUMEN BASH

LO BÁSICO

bash es una shell, que es a su vez un interprete de comandos y un sistema de programación.

Un script de bash debería empezar siempre con un shebang: `#!/bin/bash`. También funciona para indicar que se usan otros intérpretes.

Para poder ejecutar un script como un programa, requiere permisos de ejecución. Se puede cambiar con **chmod**.

Para ejecutar un script hay dos opciones:

- En una shell nueva: **bash fichero** o **./fichero**
- Sobre la shell actual: **. fichero** o **source fichero**

CARACTERES ESPECIALES

bash interpreta los siguientes caracteres de forma especial:

Caracter	Tipo
<code>`" \</code>	Sustitución y expansión
<code>&</code>	Ejecutar proceso en segundo plano
<code>< > « » ' </code>	Redirección
<code>* ? [!]</code>	Sustitución
<code>\$</code>	Variable
<code>;</code>	Separa comandos en la misma línea

EXPANSIÓN Y SUSTITUCIÓN

La sustitución se realiza antes de ejecutar un comando. Si falla, lo interpreta literalmente.

El orden es:

1. Redirección
2. Expansión de llaves
3. Expansión de viguillas/tildes
4. Expansión de variables
5. Sustitución de comandos
6. Expansión aritmética
7. Particionado de palabras (IFS)
8. Expansión de nombres de ficheros

REDIRECCIONES

Descriptores por defecto: 0 (stdin), 1 (stdout), 2 (stderr)

Caracter	Redirige
<code>cmd < file</code>	Entrada estandar de fichero
<code>cmd « etiq</code>	Entrada estandar hasta encontrar línea con sólo la etiqueta
<code>cmd > / » file</code>	Salida estandar del comando sobreescribiendo/concatenandola al fichero
<code>cmd 2> file</code>	Salida del descriptor al fichero
<code>cmd 2>&1</code>	Salida de error a salida estandar
<code>cmd &> file</code>	Salida estandar y de error a fichero
<code>cmd1 cmd2</code>	Salida estandar de cmd1 a entrada estandar de cmd2

EXPANSIÓN DE LLAVES, VIGUILLAS Y ARITMÉTICA

Las llaves permiten generar cadenas arbitrarias sin tener en cuenta los ficheros existentes

Ej: `echo a{d,c,b}e -> ade ace abe`

La viguilla se sustituye por el path al directorio home del usuario (por defecto el actual)

`$(())` y `$()` permiten evaluar expresiones aritméticas, similares a C

Ej: `$((4+11)/3) -> 5`

EXPANSIÓN DE VARIABLES

\$	Valor
<code>\${ }</code>	Permite acceder a una variable
<code>\$(cmd) 'cmd'</code>	Es sustituido por la salida estandar del comando al ejecutarse
<code>\${variable:-valor}</code>	Muestra un valor alternativo si la variable no tiene valor asignado
<code>\${variable:=valor}</code>	Asigna el valor a la variable si la variable no tiene valor asignado
<code>\${variable:?mensaje}</code>	Muestra un mensaje de error si la variable no tiene valor asignado
<code>\${variable:+valor}</code>	Reemplaza el valor de la variable con el dado solo si ya tiene valor

EXPANSIÓN DE NOMBRES

Caracter	Acción
<code>'...'</code>	Ignora todos los caracteres especiales
<code>"..."</code>	Ignora caracteres especiales salvo <code>\$</code> , <code>'</code> y <code>\</code>
<code>\</code>	Caracter de escape. Por si solo, indica continuación en siguiente línea

Comodín	Significado
<code>*</code>	0 o más caracteres
<code>?</code>	1 carácter cualquiera
<code>[]</code>	Alguno de los siguientes caracteres
<code>[!]</code>	Cualquier caracter salvo los siguientes
<code>[0-9]</code> , <code>[a-z]</code>	Sustitución por rango (orden y locale importan)

Rangos útiles:

- `[:lower:]`, `[:upper:]`: Solo minusculas/mayusculas
- `[:alpha:]`, `[:digit:]`: Caracteres alfabeticos/numéricos

CONDICIONALES

```
if [ expr ]
then
    cmd1
elif [ expr ]
then
    cmd2
else
    cmd3
fi
```

Parametro	Descripción
<code>-z str</code>	Cadena vacía
<code>-n str</code>	Cadena no vacía
<code>str1 = str2</code>	Cadenas idénticas
<code>str1 != str2</code>	Cadenas distinta
<code>str</code>	Cadena no nula
<code>int1 -eq int2</code>	Enteros iguales
<code>int1 -ne int2</code>	Enteros distintos
<code>int1 -gt int2</code>	int1 mayor estricto que int2
<code>int1 -ge int2</code>	int1 mayor o igual que int2

Parametro	Descripción
int1 -lt int2	int1 menor estricto que int2
int1 -le int2	int1 menor o igual que int2
-r file	Fichero legible
-w file	Fichero escribible
-x file	Fichero ejecutable
-f file	Fichero es regular
-d file	Fichero es directorio
-h file	Fichero es enlace simbólico
-c	Fichero es dispositivo de caracteres
-b	Fichero es dispositivo de bloque
-p	Fichero es tubería con nombre
-u	Fichero setuid-izado
-g	Fichero setgid-izado
-k	Fichero con sticky bit
-s	Fichero no vacío
!	NOT
-a	AND
-o	OR
(expr)	Agrupación (paréntesis entre comillas o escapados)

Además:

- `cmd1 && cmd2`: Ejecuta `cmd2` si `cmd1` tiene éxito
- `cmd1 || cmd2`: Ejecuta `cmd2` si `cmd1` fracasa

```
case variable in
    pattern 1 )
        cmd1
        ...
    ;;
    pattern 2 )
        cmd2
        ...
    ;;
    * )
        cmd3
        ...
    ;;
esac
```

Los patrones son comodines. | actúa como OR

BUCLES

```
for variable in str1 str2 ... strn
do
    cmd
    ...
done

while cond
do
    cmd
    ...
done

until cond
do
    cmd
```

```
...
done
```

Además:

- **break** [n] sale inmediatamente de 1/n bucles añadidos
- **continue** pasa a la siguiente iteración sin terminar la actual
- \$* y @\$ deben ir entrecomillados
- Los bucles son comandos y pueden ser redirigidos
- Se puede cambiar el IFS (por defecto espacio) localmente en el bucle:
 - while IFS=“,” , cond ...

FUNCIONES

```
foo() {
    cmd
    ...
}

function bar {
    cmd
    ...
}
```

```
foo a1 a2 a3
```

Se diferencian de las funciones de lenguajes de programación en que:

- No tienen control de número y tipo de parámetros: se acceden de la misma manera que los parámetros del script
- Devuelven valor de estado de éxito/fracaso con **return** [n], no un resultado de operación

Se puede declarar una variable como local con **local**. Así no sobrescribe variables globales/otras locales del mismo nombre

VARIABLES

VARIABLES DEL ENTORNO

Variable	Contenido
\$PATH	Rutas a programas
\$HOME	Directorio home del usuario
\$PWD	Directorio actual
\$SHELL \$TERM	Nombre de shell/terminal
\$UID \$EUID	Id del usuario (actual/efectivo)
\$USER	Nombre del usuario actual
\$PS0, \$PS1	Prompt (símbolo en la terminal)
\$DISPLAY	Numero de pantalla (Xorg)
\$MANPATH	Ruta a paginas de manual
\$IFS	Separador de campos actual

VARIABLES PREDEFINIDAS

Variable	Contenido
\$0	Nombre del script
\$1 ... \$9, \${10} ...	Parámetros
\$#	Número de parámetros
\$*	Todos los parámetros como secuencia
@\$	Todos los parámetros explícitamente separados
\$_	Estado de salida del último comando
\$\$	PID del shell

Variable	Contenido
\$!	PID último proceso segundo plano enviado

OTRAS COSAS ÚTILES

Cosa	Descripción
!!	Último comando ejecutado
!n	n-esimo comando ejecutado
!string	Primer comando que empieza por string
!?string	Primer comando que contiene string
CTRL + C	Mata proceso en primer plano
CTRL + D	Señalo EOT. Sale del shell o cierra escritura en stdin
Depuración	Con bash -x, -o xtrace o set +-x

Optimización:

- Usar aritmética de shell antes que **expr**
- Usar \$(< file) antes que \$(cat file)
- Mejor usar cadenas de comandos que preprocesen la salida de un fichero a procesarlas paso a paso manualmente en un bucle
- Evitar comandos innecesarios: un comando que hace dos cosas a la vez es mejor que dos que hacen una cada
- Primero hazlo bien, luego optimiza

COMANDOS ÚTILES

alias/unalias nombre=cadena

Crea/elimina un nombre alternativo para una cadena. Por defecto los muestra

bash [opciones] ... [cadena comandos | fichero]

Bourne-Again Shell. Lanza una nueva sesión de shell, interactiva o con una serie de comandos a ejecutar.

- -c: Lee comandos despues de opciones de lanzamiento
- -i: Interactivo
- -l: Actua como login shell
- -x: Modo depuración
- -v: Modo verboso

bg/fg PID

Manda proceso a segundo/primer plano

cat [opciones] ... [fichero]

Saca por salida estandar contenidos de un fichero (defecto entrada estandar).

- -b: Numera salida

chmod [opciones] ... modo ... fichero ...

Cambia los permisos de un fichero/directorio. Acepta varios formatos: textual (ugo+-rwx) y octal (XXXX, X <- 0-7 (r=4 w=2 x=1), tipo/U/G/O)

- -R: Recursivo

cp [opciones] ... origen ... destino

Copia archivos

- *-f*: Fuerza la copia. Usar con cuidado
- *-i*: Interactivo, pregunta antes de sobrescribir
- *-n*: No sobrescribir
- *-r*: Recursivo, para subdirectorios
- *-u*: Copia solo si origen es posterior a destino
- *-v*: Modo verboso

cut opciones ... [fichero] ...

Corta partes de cada línea de uno o varios ficheros, o de la entrada estandar.

- *-b*: Corta los N-M bytes
- *-c*: Corta los N-M caracteres
- *-d*: Cambia delimitador (tabulador por defecto)
- *-f*: Corta campos según delimitador
- *-s*: Solo devuelve líneas con delimitadores
- *-complement*: Corta el complementario a lo descrito
- *output-delimiter=*: Cambia delimitador de salida

date [opciones] [+formato]

Muestra la fecha y hora actual (según locale)

- *-d*: Muestra hora de cadena proporcionada
- *-f*: Muestra hora de cada línea de fichero proporcionado
- *-s*: Cambia la hora a la proporcionada
- *-u*: Fecha/hora UTC en vez de hora local

Formato:

- *%d %m %y %Y*: dd mm aa aaaa
- *%H %M %S*: hh mm ss
- *%D*: mm/dd/aa
- *%F*: aaaa-mm-dd
- *%R*: hh:mm
- *%T*: hh:mm:ss
- *%a/A %b/B*: Nombre del día/mes abreviado/no abreviado
- *%j*: Día del año
- *%u*: Día de la semana
- *%U*: Semana del año
- *%s*: Segundos desde *epoch*

echo [opciones]

Muestra texto en salida estandar

- *-n*: No añade \n al final
- *-e*: Interpreta caracteres de escape (\)

env [opciones] ... [comando [args] ...]

Por defecto muestra variables de entorno (**=printenv**). Permite ejecutar un programa en un entorno alterado

exec [comando [parametros]]

Sustituye shell por el comando. Si no se especifica comando, se redirige entrada/salida del propio shell

exit [n]

Salida de shell. Devuelve un entero entre 0 (ejecución correcta) y 255

expand [opciones] ... [fichero] ...

Convierte tabulaciones en espacios

- *-i*: Solo al inicio
- *-t*: Número de espacios (defecto 8)

export name[=**nuevo**]

Exporta variable para que pueda ser accedida por shells hijos nuevos, dándole un nuevo valor si se indica.

expr {expresión | opción}

calcula una expresión aritmética con enteros. Usar solo si shell es sh.

find [directorio] ... [expresion]

Busca en una serie de ficheros

Opciones:

- *-daystart*: Tiempo desde ahora
- *-depth*: Procesar contenido antes que directorio
- *-max/mindepth*: Profundidad máxima/mínima
- *-xdev*: No explorar otras particiones

Tests:

- *amin -cmin -mmin*: Último acceso/cambio estado/modificación en minutos
- *-atime -ctime -mtime*: " en días
- *-name -iname -regex*: Nombre fichero diferenciando/no diferenciado mayúsculas/expresión regular
- *-perm*: Permisos
- *-size*: Tamaño

Acciones:

-exec -ok: Ejecuta comando (preguntando)

fmt [-ancho] [opciones] ... [fichero] ...

Reformatea párrafo de fichero/entrada estandar a ancho líneas.

- *-c*: Mantiene indentación de primeras dos líneas
- *-p*: Reformatea solo líneas que empiezan con la cadena dada
- *-s*: Divide líneas largas sin rellenar
- *-t*: Primera línea con indentación diferente
- *-u*: Espacio entre palabras, dos entre oraciones.
- *-w*: Igual a -
- *-g*: Objetivo de porcentaje de anchura

grep/egrep/fgrep [opciones] ... **expr** [fichero] ...

Devuelve líneas que cumplan con una expresión regular (extendida/de un fichero)

- *-c*: Número de líneas que cumplen
- *-l*: Devuelve nombres de ficheros con coincidencias
- *-v*: Devuelve líneas que no cumplen
- *-E* (solo grep): Usa ER extendidas
- *-f* (solo egrep): Obtiene ER de un fichero

NOTA: cuidado con las expresiones, hay muchos caracteres que requieren \

head [opciones] ... [fichero] ...

Muestra por defecto las primeras 10 líneas de un fichero (defecto entrada estandar) por salida estandar

- *-c*: Primeros n bytes
- *-n*: Primeras líneas

history

Muestra el historial de bash

id [opciones] ... [usuario]

Muestra nombres e ids del usuario (EUID/EGID/grupos)

- *-u*: Muestra solo EUID
- *-g*: Muestra solo EGID
- *-G*: Muestra solo GID de grupos del que es miembro
- *-n*: Muestra solo nombres

jobs [opcion] PID

Muestra comandos en segundo plano de la shell

- *-l*: Más información
- *-p*: Solo PID

join [opciones] ... fichero1 fichero2

Dados dos ficheros, une las líneas que empiezen idénticamente, concatenando las partes que difieren de cada fichero

- *-i*: Ignora mayúsculas

less/more [opciones] [fichero]

Permite navegar lo escrito por salida estandar. Usar less antes que more

ln [opciones] objetivo ... destino

Crea enlaces a ficheros y directorios

- *-b*: Copia de seguridad de archivos destino
- *-f*: Fuerza sobreescritura
- *-i*: Interactivo
- *-L*: Enlaza lo que referencian enlaces simbólicos
- *-n*: Enlaza a los propios enlaces simbólicos
- *-P*: Hacer enlaces duros a enlaces simbólicos
- *-s*: Crea enlaces simbólicos
- *-v*: Verboso

ls [opciones] ... [fichero] ...

Muestra contenido de un directorio/información de un fichero

- *-a*: No ignorar . (A para ignorar . y ..)
- *-B*: Ignorar copias de seguridad (~)
- *-d*: Listar solo directorio
- *-h*: Formato numerico legible para humanos
- *-i*: Mostrar inodo
- *-l*: Formato largo
- *-m*: Usa comas como separador
- *-p*: Directorios empiezan con /
- *-Q*: Nombres entre “comillas”
- *-r*: Invertir orden
- *-R*: Recursivo
- *-S*: Ordenar por tamaño
- *-t*: Ordenar por fecha
- *-X*: Ordenar por extensión
- *-1*: Mostrar un fichero por fila

man [sección] nombre

Devuelve la página del manual para el comando/término proporcionado

mkdir [opciones] ... nombre

Crea un directorio

- *-m*=: Asigna permisos en vez de utilizar umask
- *-p*: Crear padres si es necesario
- *-v*: Verboso

mv [opciones] ... origen ... destino

Mueve (renombra) archivos

- *-b*: Hace copia de seguridad de archivos destino existentes
- *-f*: Fuerza el movimiento. USAR CON EXTREMO CUIDADO
- *-i*: Interactivo, pregunta antes de mover
- *-n*: No sobrescribir
- *-u*: Sobreescribe solo si origen mas nuevo
- *-v*: Verboso

nl [opciones] ... [fichero] ...

Devuelve contenido del fichero con números de línea al principio de cada una

od [opciones] ... [fichero] ...

Devuelve vuelque del fichero en octal y otros formatos

- *-endian*=: Izquierda-derecha (little), derecha-izquierda(big)
- *-j*: Salta primeros n bytes
- *-N*: Solo primeros n bytes
- *-S*: Cada byte representado por una cadena de anchura n
- *-t*: Formato
 - a: Caracteres nombrados
 - o1: octal
 - c: Caracteres imprimibles
 - f: Floats
 - dI: Enteros normales
 - dL: Enteros largos
 - o2: Octal 2 bytes
 - d2: Decimal 2 bytes
 - x2: Hexadecimal 2 bytes
- *-v*: Muestra duplicados en vez de *

paste [opciones] ... [ficheros] ...

q Devuelve concatenación de ambos ficheros, línea por línea, o de entrada estandar.

- *-d*: Sustituye TAB como delimitador
- *-s*: “Apila” las concatenaciones en vez de concatenarlas al lado en la misma linea

ps [opciones] ...

Muestra procesos en ejecución

pwd

Muestra ruta actual

read [-r] variable ...

Lee de la entrada estandar y lo asigna a una variable o variables. *-r* no trata como caracter de escape

readonly name[=nuevo]

Vuelve variable a modo solo lectura, dándole opcionalmente un nuevo valor

return [n]

Salida de una función en un script con salida n

rev [fichero] ...

cat del fichero con salida invertida, caracter a caracter

rm [opciones] ... archivos ...

Elimina archivos

- *-f*: Fuerza borrado. USAR CON EXTREMO CUIDADO
- *-i*: Interactivo para cada archivo
- *-I*: Pregunta si se eliminan tres o mas archivos
- *-r*: Recursivo
- *-d*: Elimina directorios vacíos
- *-v*: Verboso

rmdir [opciones] ... directorio ...

Elimina un directorio vacío

- *-p*: Eliminar directorios padre que sean también vacíos
- *-v*: Verboso

sed [opciones] ... [script] ... [fichero] ...

Editor de flujos no interactivo.

Ejemplos de uso:

- */ER/ ->* Condición para ejecutar siguiente sentencia
- *2,9 !,\$ ->* Rango de líneas afectadas
- *s/ER/reemplazo/flag ->* Sustituye ocurrencias
- *d ->* Borra la línea que ha saltado ocurrencia
- *i a ->* Inserta antes/después de línea
- *c ->* Reemplaza línea
- *&:* Sustituye por patrón reconocido

Opciones:

- *-r*: ER extendidas
- *-e*: Añade reglas a ejecutar
- *-f*: Añade reglas de un fichero
- *-i*: Edición in-place
- *-n*: Suprime salida

Flags:

- *g*: Cambios globales en vez de a primera ocurrencia
- *p*: Imprime líneas afectadas, sortea -n
- *5*: reemplaza la n-esima aparición
- *w fich*: Escribe líneas con sustituciones a fichero

set [opciones] [argumentos] ...

Por defecto muestra variables locales del shell. También permite modificarlas.

shift [n]

Mueve todos los parámetros 1/n a la izquierda

sort [opciones] ... [fichero]

Devuelve por salida estandar el fichero o entrada estandar ordenado lexicograficamente

- *-b*: Ignora espacios al principio de línea
- *-d*: Considerar solo alfanuméricos
- *-f*: Ignorar mayúsculas
- *-i*: Considerar solo caracteres imprimibles
- *-k*: Orden según campos
- *-n*: Por valor numérico de cadena
- *-r*: Invertir resultado
- *-t*: Separador entre campos
- *-u*: Mostrar una sola vez líneas idénticas

split [opciones] ... [fichero [prefijo]]

Divide cada 1000 líneas el fichero, creando nuevos ficheros llamados prefijo_{aa}, prefijo_{ab}, ...

- *-a*: Longitud de sufijo
- *-b*: Longitud en bytes
- *-d*: Prefijos numéricos
- *-x*: Prefijos hexadecimales
- *-e*: No genera ficheros vacíos usando *-n*
- *-l*: Longitud en líneas
- *-t*: Separador
- *-u*: Sin buffer
- *-filter=*: Usa comando como filtro

tail [opciones] ... [fichero]

Muestra por defecto las últimas 10 líneas de un fichero (por defecto entrada estandar) por salida estandar

- *-c*: Últimos *n* bytes
- *-n*: Últimas *n* líneas

tac [opciones] ... [fichero] ...

Concatenación inversa línea a línea

tee [opciones] ... [fichero]

Lee de entrada estandar y escribe en salida estandar y ficheros. *-a* concatena.

test expresión

Prueba una condición. Ver apartado de condicionales. Equivalente a [expresión]

time [opciones] ... comando [argumentos] ...

Mide el tiempo de ejecución de un comando/programa. Con *-f* se pueden obtener mas datos.

tr [opciones] ... array1 [array2]

Traduce caracteres de array1 a array2 de entrada estandar

- *-c*: Usa el complemento de array1
- *-d*: Borra en vez de traducir caracteres de array1
- *-s*: Colapsa repetidos de array1
- *-t*: Trunca array1 a largo de array2

type [nombre]

Devuelve ruta del comando o si es un comando del shell

uniq [opciones] ... [entrada [salida]]

Filtra repetidos de entrada (estandar) a salida (estandar)

- *-c*: Añade al principio cuenta de repetidos
- *-d*: Devuelve solo líneas duplicadas
- *-D*: Devuelve todas las líneas duplicadas
- *-f*: Salta primeras n líneas
- *-i*: Ignora mayúsculas
- *-s*: Ignora primeros n caracteres
- *-t*: Separador de campos
- *-u*: Devuelve solo líneas únicas
- *-w*: Compara solo primeros N caracteres

wc [opciones] ... [fichero]

Cuenta número de saltos de línea, palabras y bytes de un fichero (defecto entrada estandar) y las muestra por salida estandar

- *-c*: Cuenta bytes
- *-m*: Cuenta caracteres
- *-l*: Cuenta saltos de línea
- *-L*: Devuelve longitud de la línea más larga
- *-w*: Cuenta palabras

who [opciones] ...

Muestra usuarios con sesión iniciada

- *-q*: Cuenta usuarios conectados actualmente
- *-b*: Muestra última fecha de arranque

whoami

Devuelve nombre del usuario efectivo

xargs [opciones] [comando [argumentos]]

Permite leer argumentos de entrada estandar y los pasa a otro comando

- *-a*: Lee un fichero en vez de stdin