RESUMEN-BASH-PPS.pdf

alvarocaboof

Programación Para Sistemas

2º Grado en Ingeniería Informática

Escuela Técnica Superior de Ingenieros Informáticos
Universidad Politécnica de Madrid

RESUMEN BASH PPS

1. Mandatos importantes
   - Test: `test expression` o `[ expression ]`

     Comprueba si la condición es correcta y lo devuelve en el $#

     ARGUMENTOS

- ne	Not equal
- r	Se puede leer
-w	Se puede escribir
-d	Existe y es un directorio
-f	Existe y es un archivo

1.2 Gestión de archivos

- ls:

  Muestra los archivos existentes en el escritorio

- l	Todo lo que se puede saber sobre los archivos
- al	Muestra los archivos ocultos

- mv: Mueve y renombra los archivos
- cp: Copia archivos de un directorio a otro
- cd: Accede al directorio dado
  - sin argumentos -> accede al $/HOME
  - cd .. retrocede al dir anterior
- mkdir: Crea un directorio con el nombre dado
- rmdir: Elimina el directorio
- grep: Busca una palabra en un texto
- man: Devuelve la documentación del mandato
- source: Permite acceder a variables de otros ficheros
  - también se escribe: . ./dir/file
- wc: Cuenta

- m	Caracteres
- w	Palabras
- l	líneas

- pwd: Devuelve el directorio actual de trabajo
- cat: Permite ver el código fuente de archivos
- env: Devuelve las variable de entorno
- find: Busca ficheros con características dadas
- echo: Busca ficheros con características dadas

$#	Nº de argumentos – 1 (el propio echo)
$?	Lo devuelto por el programa al OS
$@	Array de los argumentos
$num	El arguento en la posición num

2. Variables
   - Asignar un valor a una variable: `ID_PRACTICA=$1`
   -

   a. Variables especiales

`$0` Nombre del *script*.

`$9` Parámetro del *script* o función en la posición indicada (1 a 9).

`$#` Número de parámetros posicionales o argumentos.

`$*` Lista de argumentos.

`$@` Lista de argumentos. Pero `"$@"` no es una tira de caracteres, sino copia exacta de la lista de argumentos

`$$` Identificador del propio proceso *shell*.

`$?` Valor devuelto por el último mandato ejecutado.

`$!` Identificador del último proceso lanzado en segundo plano.

3. Gestión de privilegios y permisos

```
#! /bin/bash
```

Cabecera que indica que es un archivo ejecutable

Comando **chmod:** Asigna permisos de ejecución

De lectura: 'r' (read).
De escritura: 'w' (write).
De ejecución: 'x' (execute).

Propietario: 'u' (owner / user).
Grupo: 'g' (group).
Otros: 'o' (other).
Todos: 'a' (all).

Se pueden ver los permisos de un archivo con ls -l

4. Símbolos

c1|c2|c3	Ejecuta varios comandos en la misma línea
c1&&c2	C2 se ejecuta solo si C1 se ejecuta
c1| | c2	C2 se ejecuta solo si C1 no se ejecuta
"v1 v2 v3"	Definición de string[]; string[0]= "v1"
'txt'	Cuando queremos que se represente de manera literal
ls > "hola.txt"	Redirige la salida del ls al archivo hola.txt
?	Exit estatus del último comando ejecutado O un carácter en la expansión de nombres de ficheros