

Práctica 6

Sistemas Operativos

Contenido

El shell bash

La orden echo

Protección de
caracteres
especiales

Ejecución en
primer y segundo
plano

Órdenes
relacionadas con
el control de
trabajos

La orden ps

Configuración del
shell bash

Opciones del
shell

Variables del
shell

Alias

Ficheros de
arranque

Práctica 6: Programación en el shell bash Parte I

Sistemas Operativos

Grado en Ingeniería en Informática
Departamento de Ingeniería Informática

Universidad de Cádiz

Práctica 6

Sistemas Operativos

Contenido

El shell bash

La orden echo

Protección de caracteres especiales

Ejecución en primer y segundo plano

Órdenes relacionadas con el control de trabajos

La orden ps

Configuración del shell bash

Opciones del shell

Variables del shell

Alias

Ficheros de arranque

- 1 ¿Qué es un *shell*?
- 2 El *shell* bash
- 3 Cómo obtener ayuda sobre el *shell* bash
- 4 Escribir en la salida estándar: la orden echo
- 5 Protección de caracteres especiales
- 6 Control de trabajos
- 7 Configuración del *shell* bash
- 8 Ambientes de ejecución
- 9 Mandatos incorporados

Práctica 6

Sistemas Operativos

Contenido

El shell bash

La orden echo

Protección de caracteres especiales

Ejecución en primer y segundo plano

Órdenes relacionadas con el control de trabajos

La orden ps

Configuración del shell bash

Opciones del shell

Variables del shell

Alias

Ficheros de arranque

- Un *shell* es una interfaz entre el sistema y el usuario.
- El sistema GNU/Linux dispone de varios *shells*: C, Korn, Z, bash, ...
- El *shell* bash, al igual que otros, se ocupa de:
 - Es un intérprete de órdenes.
 - Es un lenguaje de programación que nos permite escribir programas llamados *scripts*. Éstos también nos permiten personalizar nuestro entorno de trabajo
- La orden `help` nos proporciona información sobre las órdenes incorporadas en el *shell*.
- Formato: `help [orden-incorporada]`

Práctica 6

Sistemas Operativos

Contenido

El shell bash

La orden echo

Protección de caracteres especiales

Ejecución en primer y segundo plano

Órdenes relacionadas con el control de trabajos

La orden ps

Configuración del shell bash

Opciones del shell

Variables del shell

Alias

Ficheros de arranque

- Orden incorporada en el *shell* que nos permite escribir en la salida estándar los argumentos que se le pasan.
- **Formato:** `echo [opciones] argumentos`
- **Ejemplos:**

```
$ echo "Introduzca opción:"
```

```
Introduzca opción:
```

```
$
```

```
$ echo -n "Introduzca opción:"
```

```
Introduzca opción: $
```

```
$ echo -e "X\nY\nZ"
```

```
X
```

```
Y
```

```
Z
```

```
$
```

Práctica 6

Sistemas Operativos

Contenido

El shell bash

La orden echo

Protección de caracteres especiales

Ejecución en primer y segundo plano

Órdenes relacionadas con el control de trabajos

La orden ps

Configuración del shell bash

Opciones del shell

Variables del shell

Alias

Ficheros de arranque

- Hay ciertos caracteres que tienen un significado especial para el *shell*: *, ?, [], ...
- En ciertas ocasiones no nos interesa que el *shell* los interprete, en estos casos debemos protegerlos:
 - Para proteger un carácter simple podemos precederlo de \
 - Para proteger un conjunto de caracteres los entrecomillamos:
 - Comillas simples '...' Protegen todos los caracteres especiales excepto las comillas simples.
 - Comillas dobles "..." Protegen todos los caracteres especiales excepto: \$, \ y '.

Ejecución en primer y segundo plano

Práctica 6

Sistemas Operativos

Contenido

El shell bash

La orden echo

Protección de caracteres especiales

Ejecución en primer y segundo plano

Órdenes relacionadas con el control de trabajos

La orden ps

Configuración del shell bash

Opciones del shell

Variables del shell

Alias

Ficheros de arranque

- Cuando ejecutamos una orden en primer plano, hasta que no termina su ejecución no podemos mandar a ejecutar otra.
- Cuando mandamos a ejecutar una orden en segundo plano, inmediatamente el *shell* nos devuelve el control de la terminal. Esto se consigue poniendo al final de la línea de órdenes el carácter `&`.
- La ejecución de órdenes en segundo plano se suele utilizar cuando éstas no necesitan entrada y tardan mucho tiempo en ejecutarse. También es útil para la ejecución de aplicaciones X.

Órdenes relacionadas con el control de trabajos

Práctica 6

Sistemas Operativos

Contenido

El shell bash

La orden echo

Protección de caracteres especiales

Ejecución en primer y segundo plano

Órdenes relacionadas con el control de trabajos

La orden ps

Configuración del shell bash

Opciones del shell

Variables del shell

Alias

Ficheros de arranque

- El sistema asigna a cada proceso que se crea un identificador de proceso (PID).
- El *shell* asigna a cada trabajo que se ejecuta en segundo plano un identificador de trabajo. Este nos va a servir para referirnos al trabajo si queremos hacer alguna operación con él.
- La orden `jobs` nos muestra los trabajos junto con sus identificadores.
- La orden `bg` permite pasar un trabajo que se está ejecutando en primer plano a segundo plano:

Órdenes relacionadas con el control de trabajos (cont.)

Práctica 6

Sistemas Operativos

Contenido

El shell bash

La orden echo

Protección de caracteres especiales

Ejecución en primer y segundo plano

Órdenes relacionadas con el control de trabajos

La orden ps

Configuración del shell bash

Opciones del shell

Variables del shell

Alias

Ficheros de arranque

- Si queremos pasar un proceso que se está ejecutando en primer plano a segundo plano, tendremos que:
 - Parar su ejecución con CTRL-Z.
 - Pasarlo a segundo plano con la orden `bg` seguida del identificador.
- La orden `kill` permite enviar una señal a un proceso.

Práctica 6

Sistemas Operativos

Contenido

El shell bash

La orden echo

Protección de caracteres especiales

Ejecución en primer y segundo plano

Órdenes relacionadas con el control de trabajos

La orden ps

Configuración del shell bash

Opciones del shell

Variables del shell

Alias

Ficheros de arranque

- Permite obtener información sobre todos los procesos que están ejecutándose en el sistema, tanto los que ha mandado a ejecutar el usuario como los demás procesos que se estén ejecutando en el sistema.
- Formato: `ps [opciones]`

Práctica 6

Sistemas Operativos

Contenido

El shell bash

La orden echo

Protección de caracteres especiales

Ejecución en primer y segundo plano

Órdenes relacionadas con el control de trabajos

La orden ps

Configuración del shell bash

Opciones del shell

Variables del shell

Alias

Ficheros de arranque

- El *shell* bash proporciona diversas características que permiten configurarlo:
 - Opciones del *shell*
 - Variables del *shell*
 - Alias
 - Ficheros de arranque

Práctica 6

Sistemas Operativos

Contenido

El shell bash

La orden echo

Protección de caracteres especiales

Ejecución en primer y segundo plano

Órdenes relacionadas con el control de trabajos

La orden ps

Configuración del shell bash

Opciones del shell

Variables del shell

Alias

Ficheros de arranque

- Permiten modificar el comportamiento del *shell* bash.
- Para establecer una opción: **set -o opción**.
- Para desactivar una opción: **set +o opción**.
- Para ver qué opciones están establecidas y cuáles no: **set +o** o bien **set -o**.
- Algunas opciones:

Opción	Descripción
emacs	En la edición de la línea de órdenes se utilizan las teclas correspondientes del editor emacs.
noclobber	No se reescribirá un fichero existente con un operador de redirección. Para reescribir el fichero habrá que especificar > .
noglob	Desactiva la expansión de nombres de ficheros.
privileged	Se activa siempre que el uid(gid) real no sea igual que el uid(gid) efectivo.
vi	Usa teclas como las del editor vi para editar la línea de órdenes.

Práctica 6

Sistemas Operativos

Contenido

El shell bash

La orden echo

Protección de caracteres especiales

Ejecución en primer y segundo plano

Órdenes relacionadas con el control de trabajos

La orden ps

Configuración del shell bash

Opciones del shell

Variables del shell

Alias

Ficheros de arranque

- El *shell* bash utiliza una serie de variables que podemos dividir en 2 grupos:
 - Variables que el *shell* establece y el usuario no debe modificar.
 - Variables que el usuario puede modificar.
- Para establecer una variable: **VARIABLE=valor**.
- Para ver el valor de una variable: **echo \$VARIABLE**.
- Para ver todas las variables: **set**.
- Algunas de las variables establecidas por el *shell*:

Variable	Descripción
PWD	Directorio de trabajo actual.
OLDPWD	Directorio de trabajo previo.
UID	Su valor es el identificador del usuario.
EUID	Su valor es el identificador efectivo del usuario.

Práctica 6

Sistemas Operativos

Contenido

El shell bash

La orden echo

Protección de caracteres especiales

Ejecución en primer y segundo plano

Órdenes relacionadas con el control de trabajos

La orden ps

Configuración del shell bash

Opciones del shell

Variables del shell

Alias

Ficheros de arranque

- Algunas variables que el usuario puede modificar:

Variable	Descripción
PATH	Lista de directorios donde el <i>shell</i> va a buscar las órdenes que le demos cuando no empiecen por /. En esta lista los directorios van separados por el carácter :.
HOME	Contiene el camino del directorio de entrada del usuario. El valor de HOME es el argumento que usa cd por omisión.
PS1	Indicador primario. Por omisión toma el valor bash-n:\$, siendo n el número de la versión del <i>shell</i> .
PS2	Indicador secundario. Si presionamos Return antes de introducir una línea de órdenes completa, nos aparece el indicador secundario. Por omisión es >.

Práctica 6

Sistemas Operativos

Contenido

El shell bash

La orden echo

Protección de caracteres especiales

Ejecución en primer y segundo plano

Órdenes relacionadas con el control de trabajos

La orden ps

Configuración del shell bash

Opciones del shell

Variables del shell

Alias

Ficheros de arranque

- Permiten dar nombres alternativos a las órdenes.
- Son útiles para dar un nombre corto a una línea de órdenes larga.
- Para ver la lista de alias definidos: **alias**.
- Para establecer un nuevo alias: **alias nombre=valor**.
- Para ver el valor de un alias determinado: **alias nombre**.
- Para desestablecer un alias: **unalias nombre**.
- Ejemplo: **alias ll='ls -l'**

Práctica 6

Sistemas Operativos

Contenido

El shell bash

La orden echo

Protección de caracteres especiales

Ejecución en primer y segundo plano

Órdenes relacionadas con el control de trabajos

La orden ps

Configuración del shell bash

Opciones del shell

Variables del shell

Alias

Ficheros de arranque

- Los ficheros de arranque nos permiten configurar las características del *shell* que queramos tener siempre disponibles: opciones, variables, alias, etc.
- El *shell* bash utiliza distintos scripts de arranque:
 - 1 En primer lugar siempre lee `/etc/profile`, fichero de arranque genérico creado por el administrador del sistema.
 - 2 Si estamos arrancando en modo texto: Busca en el directorio de casa y lee el primero de los siguientes ficheros: `~/.bash_profile`, `~/.bash_login` y `~/.profile`.
 - 3 Si arrancamos en modo gráfico y llamamos a una terminal el *shell* lee el fichero `~/.bashrc`.
- Al terminar la sesión el *shell* lee el fichero `~/.bash_logout`, si existe.

Práctica 6

Sistemas Operativos

Contenido

El shell bash

La orden echo

Protección de caracteres especiales

Ejecución en primer y segundo plano

Órdenes relacionadas con el control de trabajos

La orden ps

Configuración del shell bash

Opciones del shell

Variables del shell

Alias

Ficheros de arranque

- Cuando mandamos a ejecutar cualquier programa desde el *shell*, se crea un nuevo proceso (proceso hijo del *shell*).
- Cada proceso tiene su propio ambiente de ejecución, este ambiente está formado con un conjunto de valores de variables, privilegios y recursos.
- Los procesos hijos heredan ciertas características del ambiente del proceso padre.
- La orden **export** permite exportar variables y funciones.

No heredadas	Alias Opciones
Heredadas	Derechos de acceso a los ficheros, directorios, etc. Ficheros abiertos. Límites de recursos. Ej.: la cantidad de memoria principal que el proceso puede usar. La respuesta del padre a señales.
Heredadas, si se exportan	Funciones Variables

Práctica 6

Sistemas Operativos

Contenido

El shell bash

La orden echo

Protección de caracteres especiales

Ejecución en primer y segundo plano

Órdenes relacionadas con el control de trabajos

La orden ps

Configuración del shell bash

Opciones del shell

Variables del shell

Alias

Ficheros de arranque

- Un *dot script* es un script que se ejecuta en el ambiente del proceso padre, es decir, no se crea un proceso nuevo para su ejecución.
- Esto le permite compartir todas las características del ambiente del *shell*.
- Para ejecutar un script como un *dot script*: `. mi_script`.

Práctica 6

Sistemas Operativos

Contenido

El shell bash

La orden echo

Protección de caracteres especiales

Ejecución en primer y segundo plano

Órdenes relacionadas con el control de trabajos

La orden ps

Configuración del shell bash

Opciones del shell

Variables del shell

Alias

Ficheros de arranque

- Un *subshell* es una copia del *shell* que se ejecuta como un proceso hijo de éste.
- Hay que tener en cuenta que los cambios que realizan los procesos hijos en el ambiente de ejecución no lo heredan nunca los procesos padres.
- Ejemplos:
 - 1

```
$ A=hola  
$ echo $A  
hola
```
 - 2

```
$ (A=hola) Al encerrar entre paréntesis la orden, se  
ejecuta en un subshell.  
$ echo $A  
¿Cuánto valdrá A?
```

Práctica 6

Sistemas Operativos

Contenido

El shell bash

La orden echo

Protección de caracteres especiales

Ejecución en primer y segundo plano

Órdenes relacionadas con el control de trabajos

La orden ps

Configuración del shell bash

Opciones del shell

Variables del shell

Alias

Ficheros de arranque

- Algunas de la órdenes que conocemos no son programas en el sentido usual, sino que están incorporadas en el *shell*.
- Al estar incorporadas en el propio *shell*, al ejecutarlas no se crea un proceso hijo.
- ¿Por qué existen las órdenes incorporadas? Hay ciertas órdenes que si no fueran de este tipo no funcionarían.
- Ejemplo: La orden `cd`
Esta orden nos permite cambiar de directorio modificando la variable del *shell* `PWD`, es decir cuando damos `pwd` `nuevo_directorio` esta orden da el valor `nuevo_directorio` a la variable `PWD`. ¿Qué ocurriría si `cd` fuese una orden normal, al terminar de ejecutarse?
- Son órdenes incorporadas en el *shell*: `.`, `cd`, `alias`, `unalias`, `bg`, `fg`, `jobs`, `echo`, `umask`, `export`, `pwd`, `help`, ...