

Sistemas Operativos

Práctica 1

Sesión 1

Manejo básico del shell bash de Unix

Curso 2005/2006

Arrancar el Equipo

1. boot: pulsar <Intro>
2. Fedora CORE
Nombre de Usuario: lnx
Password: lnx
3. Red-Hat Linux 7.2-Aulas
4. Localhost login: alumno
Password: alumno

Apagar: halt -p

Bibliografía

- **Amir Afzal, Introducción a UNIX: un enfoque práctico, Prentice-Hall, 1997.**
- **Cameron Newham, Bill Rosenblatt, Learning the bash shell, , 3ª edición, O'Reilly, Marzo 2005.**
- Manual de Referencia de Bash online:
http://www.faqs.org/docs/bashman/bashref_toc.html
- Tarjeta de referencia de comandos UNIX:
<http://www.indiana.edu/~uitspubs/b017/b017.pdf>

Introducción a la shell UNIX

- Interfaz de usuario basada en caracteres con el sistema operativo tipo UNIX (Linux, AIX, SunOS, etc.)
- Interacción con el usuario por línea de comandos
- Salida de los programas basada en texto
- Traduce las líneas de comandos en instrucciones del sistema operativo

Introducción a la shell UNIX

- La shell NO es UNIX, sólo una interfaz con él
- UNIX es uno de los primeros SSOO en hacer la interfaz de usuario independiente del SO
- Alcance:
 - Interfaz de usuario
 - Entorno de programación. Shell scripting.

Introducción a la shell UNIX

- Un poco de historia:
 - Primera shell relevante fue la shell Bourne (en honor a su inventor Steven Bourne) incluida en la primera versión más popular de UNIX, la versión 7, en el años 1979.
 - Alternativas ampliamente utilizadas: la shell C que reemsamblaba funciones del lenguaje C en forma de comandos de la shell. Tenía características únicas como el control de trabajos.
 - La shell más popular en los últimos tiempos es la shell Korn. Es una solución comercial y adopta lo mejor de las shells Bourne y C agregando algunas prestaciones propias

Bourne Again Shell (BASH)

- Bourne Again Shell (BASH) fue creada para su uso en el proyecto GNU.
- Se lanzó oficialmente el 10 de Enero de 1988 y su autor original fue Brian Fox que la mejoró hasta 1993. Chet Ramey es el actual responsable de su mantenimiento.
- Es de libre distribución, código abierto y es considerada la shell estándar
- Estudiaremos la versión 3.0

Bourne Again Shell (BASH)

- Prestaciones más relevantes:
 - Edición de la línea de comandos
 - Control de trabajos
 - Personalización de la shell
 - Completo entorno de programación
- Bash tiene como objetivo ser una implementación conforme de la especificación de Herramientas y Shell IEEE POSIX (IEEE Working Group 1003.2).

Bash - arranque

- Programa login mira la shell definida en el fichero /etc/passwd

```
[user@localhost ~]# cat /etc/passwd
```

- Encontrar vuestra entrada de usuario

```
carlos:x:500:500:~/home/carlos:/bin/bash
```

- **Posibles shell en el fichero /etc/shells**

```
[user@localhost ~]# cat /etc/shells
```

Bash - arranque

- **Ficheros que lee bash cuando se arranca :**
 1. /etc/profile
 2. ~/.bashrc, ~/.bash_profile, ~/.bash_login, and ~/.profile
- **Más información sobre el bash:**

```
[user@localhost ~]# man bash
```

Bash - comandos

- Determinar la shell que estamos utilizando

```
[user@localhost ~]# echo $SHELL
```

- Cambiar a otra shell

```
[user@localhost ~]# chsh <path del shell>
```

- Cambiar a otro terminal

```
Alt + F $\Phi$      $\Phi=1,2,\dots$ 
```

- Sintaxis

```
comando [opciones] [argumentos]
```

Bash - comandos

- Básicos

`exit`: termina el shell actual

`date`: devuelve hora y fecha del sistema

`who`: devuelve los usuarios conectados

`whoami`: imprime el usuario efectivo

`cal`: muestra el calendario del mes actual

`locate [nombre de fichero]`: búsqueda rápida de ficheros

`updatedb`: actualiza BBDD que utiliza `locate`

`which [nombre de fichero]`: muestra el path completo de comandos de la shell

`clear`: resetea la pantalla del terminal

PRACTIQUEMOS UN POCO

Bash - comandos

- Básicos

man: accede al manual

```
[user@localhost ~]# man bash
```

alias/unalias: macros para comandos

```
[user@localhost ~]# ls
```

```
[user@localhost ~]# ls -l
```

```
[user@localhost ~]# alias ls='ls -l'
```

```
[user@localhost ~]# ls
```

```
[user@localhost ~]# unalias ls
```

```
[user@localhost ~]# ls
```

Bash - comandos

- **Manejo del entorno**

`set/unset`: establece/elimina valor a variable de entorno

`echo $<VARIABLE_ENTORNO>`

```
[user@localhost ~]# set
```

`export <VARIABLE_ENTORNO>=VALOR`: exporta la variable al entorno para que sea utilizada por los comandos ejecutados desde ese momento

```
[user@localhost ~]# export PRUEBA=prueba
```

```
[user@localhost ~]# echo $PRUEBA
```

```
[user@localhost ~]# unset PRUEBA
```

```
[user@local, ost ~]# echo $PRUEBA
```

Bash - comandos

- Variables del entorno:
 - PATH, BASH, HISTSIZE, HISTFILESIZE, HISTFILE, HOME, HOSTNAME, LANG, USER, UID

```
[user@localhost ~]# echo $PATH
```

```
[user@localhost ~]# echo $BASH
```

Bash - comandos

- Manejo de Directorios y Ficheros

`touch`: permite cambiar la fecha de un archivo (y crear ficheros vacíos)

```
[user@localhost ~]# cd
```

```
[user@localhost ~]# touch fichero.prueba
```

`cp`: copiar un fichero en otro o ficheros en un directorio

```
cp [-R] [directorio/fichero origen]
    [directorio/fichero destino]
```

`cd`: navegar por los directorios

```
cd [directorio] cd- cd~
```

`mkdir`: crear directorios

```
mkdir [-p] [directorio]
```

Bash - comandos

- Manejo de Directorios y Ficheros

`rmdir`: elimina un directorio (debe estar vacío)

```
rmdir [nombre_directorio]
```

`rm`: elimina un fichero

```
rm [-r] [-f] nombre_fichero
```

Editor de textos JOE

[user@localhost ~]# joe <nombre de fichero>

- ^kh activa el menú de ayuda
- ^kb marcar el comienzo de un bloque
- ^kk marcar el final de un bloque
- ^kc copiar un bloque
- ^km mover un bloque
- ^ky borrar un bloque
- ^kd guardar fichero
- ^kx salir guardando
- ^c salir sin guardar

PRACTIQUEMOS UN POCO ...

Bash - comandos

Crear en el directorio \$HOME la siguiente estructura de archivos. Utilizar el comando mkdir (comprobar la utilidad de la opción -p) para crear los directorios y el editor de textos JOE para crear los ficheros dentro la estructura de directorios ...

Bash - comandos

```
$HOME/fichero.prueba
```

```
  /practicas
```

```
  /p1
```

```
  /src/p1.c
```

```
  .p1.config
```

```
  /bin/p1
```

```
  /p2
```

```
  /src/p2.c
```

```
  /bin/p2
```

```
  /doc/p2.doc
```

```
  /para_borrar1/borrar.1
```

```
  /para_borrar2/borrar.2.1
```

```
  borrar.2.2
```

Bash - comandos

`ls`: listar el contenido de un directorio

```
[user@localhost ~]# cd $HOME
```

```
[user@localhost ~]# cd practicas/p1
```

```
[user@localhost ~]# ls -l
```

```
[user@localhost ~]# ls -la
```

`mv`: mover un fichero desde una ubicación de la estructura de directorios a otra

```
[user@localhost ~]# cd $HOME
```

```
[user@localhost ~]# cp fichero.prueba fichero.prueba.2
```

```
[user@localhost ~]# mv practicas practicas2005_06
```

```
[user@localhost ~]# mv -i fichero.prueba.2 fichero.prueba
```

Bash - comandos

`pwd`: indica el camino absoluto del directorio en el que nos encontramos

```
[user@localhost ~]# cd $HOME/practicas2005_06/p2  
[user@localhost ~]# pwd
```

`cd` (practicar con el tabulador para autocompletar)

```
[user@localhost ~]# cd  
[user@localhost ~]# cd practicas2005_2006/p1  
[user@localhost ~]# pwd  
[user@localhost ~]# cd ~  
[user@localhost ~]# pwd  
[user@localhost ~]# cd  
[user@localhost ~]# pwd
```

Bash - comandos

`rm, rmdir`

```
[user@localhost ~]# pwd
```

```
[user@localhost ~]# rmdir practicas2005_2006/p1 ---> ERROR
```

```
[user@localhost ~]# rm practicas2005_2006/para_borrar1/borrar.1
```

```
[user@localhost ~]# rmdir practicas2005_2006/para_borrar1/borrar.1
```

```
[user@localhost ~]# rm -f practicas2005_2006/para_borrar1/borrar.1
```

```
[user@localhost ~]# rm practicas2005_2006/para_borrar1/borrar.1
```

`cat`: concatena ficheros (o la entrada estándar) y los saca por la salida estándar

```
[user@localhost ~]# cat /etc/passwd
```

Bash - comandos

`more, less, head y tail`: paginadores de texto

```
[user@localhost ~]# head -2 /etc/passwd
```

```
[user@localhost ~]# tail -2 /etc/passwd
```

```
[user@localhost ~]# more /etc/passwd
```

```
[user@localhost ~]# less /etc/passwd
```

`du`: estima el uso de espacio de ficheros

```
du [-h] [--max-depth] [fichero]
```

```
[user@localhost ~]# cd
```

```
[user@localhost ~]# du
```

```
[user@localhost ~]# du -h
```

```
[user@localhost ~]# du -h --max-depth=1
```