

Qué es BASH?

- Shell interactiva de comandos (ksh + csh)
- Historia
 - ◆ UNIX™: Dennis Ritchie y Ken Thompson (AT&T)
 - ◆ S.R. Bourne... *SHELL*
 - ◆ Interprete de comandos de GNU
 - ◆ Bourne Again Shell
- IEEE posix Shell
- Versión actual: 3.00.14(1)



Definiciones (1/4)

- Espacio (blank)
 - ♦ espacio o tab
- Palabra (word o token)
 - ♦ Secuencia de caracteres considerado como una única unidad
 - ★ snoopy
 - ★ m@v3^R|z!
 - ★ `[\033[0;30\]`



Definiciones (2/4)

- Nombre (name o identifier)
 - ♦ Caracteres alfanuméricos y underscores
 - ★ hola_mundo
 - ★ _h4ck3r
- Metacaracter
 - ♦ Separador de palabras
 - ★ | & ; () < > [space] , [tab]



Definiciones (3/4)

- Operador de control
 - ◆ Es un token que efectúa una función de control
 - ★ | y ||
 - ★ & y &&
 - ★ ; y ;;
 - ★ ()
 - ★ <newline>



Definiciones (4/4)

- Línea de Comandos (Pipelines)

- ♦ Secuencia de uno o más comandos, separados por “|”

- ★ `ls | less`

- ★ `cat archivo | grep algo`

- Listas

- ♦ Secuencia de uno o más pipelines, separados por operadores de control

- ★ `Comando1 || comando2`

- ★ `Comando1 && comando2`



Archivos Importantes

- /etc/profile
- ~/.bash_profile
- ~/.bashrc
- ~/.inputrc
- ~/.bash_aliases
- ~/.bash_logout



Algunos comandos básicos

- cd
- ls
- mkdir
- rm
- cp
- mv
- pwd
- cat
- tail
- head
- less
- grep
- sort
- cut
- echo
- alias
- test
- xargs
- wc
- shopt
- find



Variables útiles (1/5)

- **\$HOME**
 - ◆ /home/mave
- **\$PWD**
 - ◆ /opt/mp3
- **\$OLDPWD**
 - ◆ /var/log
- **\$PATH**
 - ◆ /bin:/usr/bin:/usr/local/bin



Variables útiles (2/5)

- \$LANG
 - ◆ en_US.UTF-8
- \$USER
 - ◆ tupac
- \$UID
 - ◆ 501
- \$RANDOM
 - ◆ 0 - 32767



Variables útiles (3/5)

- **\$HISTFILE**
 - ◆ `$HOME/.bash_history`
- **\$HISTSIZE**
 - ◆ 500
- **\$HISTFILESIZE**
 - ◆ 500
- **\$HISTCMD**
 - ◆ 2038



Variables útiles (4/5)

- \$HOSTNAME
 - ◆ kayupi
- \$HOSTTYPE
 - ◆ i386
- \$OSTYPE
 - ◆ linux-gnu
- \$PPID
 - ◆ 11391



Variables útiles (5/5)

- `$BASH_COMMAND`
 - ◆ *El comando que está siendo o será ejecutado*
- `${BASH_VERSINFO[@]}`
 - ◆ 3 00 14 1 release i386-redhat-linux-gnu
- `$VISUAL`
 - ◆ `jed`
- `$EDITOR`
 - ◆ `/usr/bin/vim`



Ocupando Variables

- `VarX=VALOR`
 - ♦ `echo $VarX`
 - ★ VALOR

- `VarX=$HOME`
 - ♦ `echo $VarX`
 - ★ /home/mave
 - ♦ `cd $VarX`
 - ★ cd /home/mave



Declarando Variables

- **set** VARIABLE
- **export** VARIABLE
- **declare** VARIABLE

- **unset** VARIABLE



Redirecciones (1/3)

- `echo "hola que tal" > /tmp/salida.txt`
 - ◆ `cat /tmp/salida.txt`
hola que tal
- `echo "chao me voy" > /tmp/salida.txt`
 - ◆ `cat /tmp/salida.txt`
chao me voy
- `echo "mejor no" >> /tmp/salida.txt`
 - ◆ `cat /tmp/salida.txt`
chao me voy
mejor no



Redirecciones (2/3)

- `make 2> /tmp/errores`
- `make > /tmp/salida 2> /tmp/errores`
- `make &> /tmp/compilacion.txt`
- `make 2>&1`



Redirecciones (3/3)

- `ls -l /etc | more`
- `prog1 < entra.txt | prog2 | prog3 > sale.txt`
- `prog1 < entra | tee inter | prog2 > sale.txt`



Atajos de teclado (1/3)

- C-a Inicio Linea
- C-e Fin Linea
- C-w Cortar palabra atras
- C-u Cortar linea atras
- C-k Cortar linea adelante
- C-y Pegar



Atajos de teclado (2/3)

- C-l Limpiar Terminal
- C-c Cancelar
- C-d Borrar caracter
- C-s Bloquear terminal ScrollLock
- C-q Desbloquear terminal
- C-r Buscar en el Historial



Atajos de teclado (3/3)

- C-t Transponer letras
- M-t Transponer palabras
- M-c Primera mayúscula
- M-u Todas mayúsculas
- M-l Todas minúsculas
- C-d logout



Expansión e interpretación (1/6)

- $\{a,b,c\}XX\{1,.\}$
 - ♦ aXX1 aXX bXX1 bXX cXX1 cXX
- $\{5..12\}$
 - ♦ 5 6 7 8 9 10 11 12
- a^*z
 - ♦ az a123z azaz abcd...wxyz
- $a?z$
 - ♦ a1z amz aZz



Expansión e interpretación (2/6)

- `~mave`
 - ◆ `/home/mave`
- `~/algun/archivo`
 - ◆ `/home/algun/archivo`
- `~+`
 - ◆ `$PWD`
- `~-`
 - ◆ `$OLDPWD`



Expansión e interpretación (3/6)

- $\$(comando)$ equivalente a 'comando'
 - ◆ $\$(cat\ archivo)$
 - ◆ 'cat archivo'
 - ◆ $\$(<\ archivo)$
- comando1 $\$(comando2)$
 - ◆ $cd\ \$(ls\ -d\ /t^*)$



Expansión e interpretación (4/6)

- `XX=(autos botes casas dados elefantes)`
- `echo $XX`
 - ♦ autos
- `echo ${XX[2]}`
 - ♦ casas
- `echo ${#XX}`
 - ♦ 5



Expansión e interpretación (5/6)

- `XX=(autos botes casas dados elefantes)`
- `echo ${XX[1]:1}`
 - ♦ otes
- `echo ${XX[1]:1:3}`
 - ♦ ote
- `echo ${XX[1]/s}`
 - ♦ bote



Expansión e interpretación (6/6)

- `XX=(autos botes casas dados elefantes)`
- `echo ${XX[1]/s/z}`
 - ♦ botez
- `echo ${XX[@]}`
 - ♦ autos botes casas dados elefantes

