

Programación en Shell

Primera línea:
#! Intérprete

Ejemplo:
#!/bin/bash
echo Hola!!!

Programación de Sistema Operativo
Héctor Abarca A.

Ejercicio uno.sh

```
#!/bin/sh
#
for f in /etc/*
do
  if [ -d $f ]
  then
    echo $f
  fi
done
exit 0
```

Programación de Sistema Operativo
Héctor Abarca A.

¿Cómo ejecutar un script?

El script debe ser interpretado:
\$ /bin/sh ./uno.sh ó
\$./uno.sh ó
\$ sh -x ./uno.sh

O puede ser convertido en ejecutable
\$ chmod +x uno.sh
\$ chmod 755 uno.sh
\$./uno.sh

Programación de Sistema Operativo
Héctor Abarca A.

Uso de variables. dos.sh

```
#!/bin/sh
mivar="Hola"
echo $mivar
echo "$mivar"
echo `mivar`
echo \mivar

echo Ingrese un texto
read mivar

echo `mivar` equivale a $mivar
exit 0
```

Programación de Sistema Operativo
Héctor Abarca A.

Paso de argumentos. tres.sh

Ejecutar este script como:
./tres a bc d

```
#!/bin/sh
echo "Este programa se llama $0"
echo "el 2do parametro es $2"
echo "el 1er parametro es $1"
echo "todos los parametros son = $*"

echo "el directorio del usuario es $HOME"
exit 0
```

Programación de Sistema Operativo
Héctor Abarca A.

Test o []

```
if test -f hola.txt
then
....
fi
```

Equivale a...

```
if [ -f hola.txt ]
then
....
fi
```

Programación de Sistema Operativo
Héctor Abarca A.

Más test... if...

```

string1 = string2
string1 != string2
-n str           Verdadero si str != Null
-z str           Verdadero si str = Null

exp -eq exp2
exp -ne exp2
exp -gt exp2
exp -ge exp2
exp -lt exp2
exp -le exp2
! exp           V si exp es F, y vice-versa
    
```

Programación de Sistema Operativo
Héctor Abarca A.

... y más test...

```

[ -d file ]    V si file es directorio
[ -e file ]    V si file existe
[ -f file ]    V si file es regular
[ -g file ]    V si set-group-id está
[ -r file ]    V si file es "read"
[ -s file ]    V si file es != 0
[ -u file ]    V si set-user-id está
[ -w file ]    V si file es "write"
[ -x file ]    V si file es "execute"
    
```

Programación de Sistema Operativo
Héctor Abarca A.

If...then. cuatro.sh

```

#!/bin/sh

echo "Son menos de las 12am? (si o no)"
read resp

if [ $resp = "si" ]
then
    echo "Buenos Dias..."
elif [ $resp = "no" ]; then
    echo "Buenas Tardes..."
else
    echo "No entiendo su respuesta!"
    exit 1
fi
exit 0
    
```

Programación de Sistema Operativo
Héctor Abarca A.

For... cinco.sh y seis.sh

```
#!/bin/sh
for lab in hola como estas 44
do
    echo la palabra clave es: $lab
done
exit 0
```

```
#!/bin/sh
for file in $(ls *.sh); do
    wc -l $file
done
exit 0
```

Programación de Sistema Operativo
Héctor Abarca A.

While... siete.sh

```
#!/bin/sh
aa=1
while [ "$aa" -le 20 ]
do
    echo "Hola version $aa"
    aa=$((aa+1))
done
exit 0
```

Programación de Sistema Operativo
Héctor Abarca A.

Until... ocho.sh

```
#!/bin/sh
until who | grep "$1" > /dev/null
do
    sleep 60
done

echo -e \\\a
echo "***** $1 se ha logeado *****"
exit 0
```

\$. ./ocho.sh usuario

Programación de Sistema Operativo
Héctor Abarca A.

Case... nueve.sh

```
#!/bin/sh
echo "Son menos de las 12am? (si/no)"
read dia
case "$dia" in
  si | s) echo "Buenos Dias...";;
  no | n | N*) echo "Buenas Tardes...";;
  *) echo "No entiendo su respuesta!";;
esac
exit 0
```

Programación de Sistema Operativo
Héctor Abarca A.

Funciones... diez.sh

```
#!/bin/sh
si_o_no() {
  echo "Es este tu nombre $* ?"
  while true
  do
    echo -n "Ingresar si/no: "
    read resp
    case "$resp" in
      s | si ) return 0;;
      n | no ) return 1;;
      *) echo "Responde si/no"
    esac
  done
}
echo "los argumentos son $*"
if si_o_no "$1"; then echo "Hola $1"
else ; echo "Me equivoque!"
fi
exit 0
```

Programación de Sistema Operativo
Héctor Abarca A.

Varios

```
x=`expr $x + 1`

printf "%s %d\t%s" "Hola" 15 veces

set $(date)
echo El mes es $2
```

Programación de Sistema Operativo
Héctor Abarca A.

Debugging (búsqueda de los “bugs”)

```
sh -n <script>
```

- Revisa errores sintaxis, no ejecuta los comandos.

```
sh -v <script>
```

- Imprime a pantalla los comandos antes de ejecutarlos

```
sh -x
```

- Imprime a pantalla los comandos después de ejecutarlos.

Programación de Sistema Operativo
Héctor Abarca A.

awk

Awk es un lenguaje de procesamiento de datos dado un patrón de búsqueda.

```
$ cat /etc/passwd | awk -F: '{printf("El usuario %s posee el directorio %s\n", $1, $6)}'
```

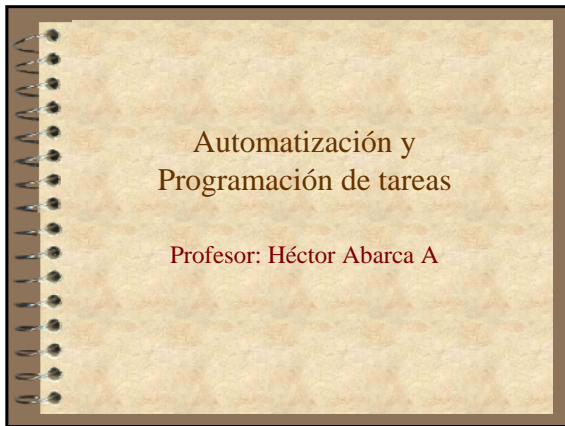
```
$ ls -l | grep drwx | awk '{printf("directorio = %s \n", $9
```

Programación de Sistema Operativo
Héctor Abarca A.

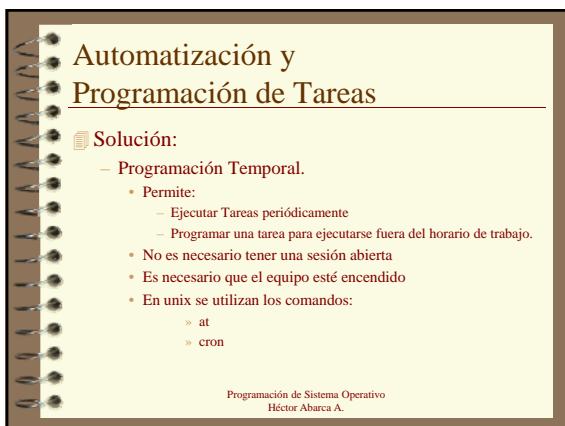
Ejercicios

- Escriba un script que dado el “uid” indique cuantas veces está logeado (who, grep, wc)
- Escriba un script llamado “lslink” que sólo liste los archivos tipo link.
- Ordene alfabéticamente el listado de usuarios de su máquina linux (sort, /etc/passwd)

Programación de Sistema Operativo
Héctor Abarca A.







Automatización y Programación de Tareas

Comando **at**

- Programar una tarea para un momento concreto.
- Son ejecutadas por el demonio atd
- Funciona con colas de tareas
- Puede condicionarse la ejecución a la carga del sistema
- at → programar una tarea
- atq → consultar tareas
- atrm → borrar tareas

Programación de Sistema Operativo
Héctor Abarca A.

Automatización y Programación de Tareas

Ejecución periódica **cron**

- Demonio del sistema crond
 - Consulta las tareas a ejecutar cada minuto
- Gestión de tareas con crontab
 - crontab -l → listar tareas
 - crontab -e → editar tareas
 - crontab -r → borrar tareas

Programación de Sistema Operativo
Héctor Abarca A.

Automatización y Programación de Tareas

Ejecución periódica **cron**

- Entradas de la forma
min hora día mes día _ semana tarea
- Se admiten rangos, listas y *
- Ejemplos:
 - 0 0 * * * * → todos los días a las 0:00
 - 45 12 * * * 1-5 → de lunes a viernes a las 12:45
 - 30 5 1,15 * * * → a las 5:30 los días 1 y 15 de cada mes

Programación de Sistema Operativo
Héctor Abarca A.

Automatización y Programación de Tareas

📖 **Ejecución periódica cron**

- Las tareas del sistema se almacenan en directorios especiales.
- /etc/cron.hourly → ejecución cada hora
- /etc/cron.daily → ejecución cada día
- /etc/cron.weekly → ejecución cada semana
- /etc/cron.monthly → ejecución cada mes

Programación de Sistema Operativo
Héctor Abarca A.

Ejercicios

📖 Cree un crontab que realice una copia de seguridad todos los lunes a las 2 de la mañana.

📖 Cree un crontab que reinicie la máquina los días lunes, miércoles y viernes de cada mes a la 21:30.

Programación de Sistema Operativo
Héctor Abarca A.
