

UD 01: Actividades Scripts (I)

n1.- Verdadero o falso (bash).

- Los shells de Linux forman parte del núcleo del Sistema Operativo.
Falso son programas que ejecutan el usuario
- El shell que se ejecuta por defecto cuando un usuario inicia sesión en un sistema Linux queda definido en el archivo `/etc/passwd`.
Verdadero
- Los programas escritos para bash pueden ser ejecutados sin problemas con sh.
Falso ya que sh es el intérprete más antiguo y puede ser que algunas opciones no las entienda
- El shell bash interpreta siempre el espacio en blanco como separador de órdenes, opciones o argumentos.
Verdadero que esta declarado en la variable IFS
- Una variable, en bash, definida en un proceso hijo, ya sea local o de entorno, no es visible por su proceso padre.
Verdadero

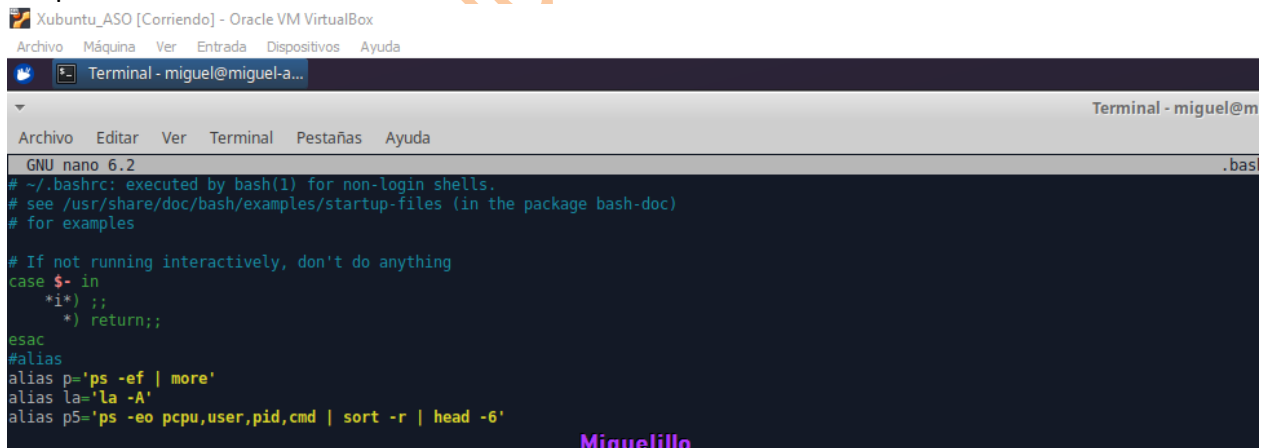
2.- Modificar el prompt principal de tal manera que aparezca el nombre de la máquina, el carácter @, el directorio actual, el carácter \$ y un espacio.

¿Si queremos tener este nuevo prompt, de forma permanente, tanto en el shell de conexión como en los posibles shells hijos que se pudieran necesitar, ¿dónde lo harías?

Tendríamos que editar el archivo `./bashrc` y añadir la línea de comando

`PS1='\h@\w$'`

3.- Crea un alias, p, que corresponda al comando `ps -ef | more`. Este alias debe ser permanente.



```

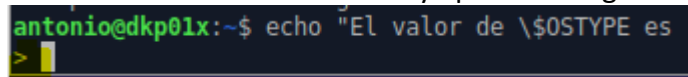
Xubuntu_ASO [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Terminal - miguel@miguel-a...
GNU nano 6.2 .bas
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
  *i*) ;;
  *) return;;
esac
#alias
alias p='ps -ef | more'
alias la='la -A'
alias p5='ps -eo pcpu,user,pid,cmd | sort -r | head -6'
Miguelillo

```

Tendríamos que editar el archivo `./bashrc` y añadir `alias 'ps -ef | more'`

4.- Mientras nos encontramos definiendo un comando, hemos tocado [INTRO] antes de haber terminado el comando y aparece lo siguiente:



```

antonio@dkp01x:~$ echo "El valor de \$OSTYPE es
>

```

¿qué significa el carácter >?

Es el prompt secundario que significa que el comando no ha sido completado en este caso faltan por cerrar las comillas, podemos ver su valor con `$PS2`

UD 01: Actividades Scripts (I)

5.- Los nombres de variables en Linux comienzan por una letra (mayúscula o minúsculas) y a continuación un número indeterminado de letras, dígitos o el carácter `_`. ¿Qué expresión regular nos permitiría comprobar la validez de la definición de una variable?

`=~ ^\[A-Za-z][A-Za-z0-9_]{1,}$`

6.- ¿Qué estamos haciendo con la siguiente expresión?

`IDENTI=$(grep antonio /etc/passwd | cut -d":" -f3)`

Puedes cambiar antonio por otro nombre

Con esta expresión declaramos el valor de la variable `IDENTI` que es igual al contenido de `/etc/passwd` que contenga el nombre de Antonio y sea el campo 3 delimitado por `:`.

7.- ¿Qué diferencia hay entre **`echo $HOME`** y **`echo '$HOME'`**?

Con `echo $HOME` nos mostrará el contenido de la variable `$HOME` en cambio con `echo '$HOME'` nos muestra literalmente `$HOME` por la comilla simple

8.- ¿Qué obtendremos al teclear `echo \\\$HOME`? ¿Por qué?

Devuelve `\HOME` ya que `"\"` se usa para mostrar caracteres literales con lo cual interpretará la primera barra como que queremos mostrar el siguiente carácter en este caso la otra barra y la tercera barra interpretará que queremos poner `$HOME` literalmente sin mostrar su valor

9.- Tenemos el siguiente script, `primero.sh`:

```
#!/bin/bash
pwd
cd /tmp
pwd
ls
```

Ese script ya tiene asignados permisos de ejecución. Se pide que digas lo que se mostrará en pantalla la siguiente secuencia de comandos, justificando tu respuesta:

```
$pwd
$primero.sh
$pwd
```

En caso que estemos en el directorio de conexión del usuario y el script se encuentre en un directorio declarado en `$PATH` cuando lo ejecutemos el comando `pwd` nos mostrará en el directorio actual en el que nos encontramos en este caso `/home/usuario`, después al ejecutar `primero.sh` se va a una Shell hija nos volverá a mostrar el directorio actual en este caso nuevamente `/home/usuario`, seguidamente nos cambiará de directorio a `/tmp` volverá a mostrar el directorio actual `/tmp`, por último hará un `ls` y nos mostrará el contenido de `/tmp` y terminará el script, por último hacemos un `pwd` en el Shell principal que nos mostrará el directorio en el que nos encontramos `pwd` en este caso `/home/usuario`.

Resultado de manera más gráfica:

```
$pwd
/home/usuario
$primero.sh
```

UD 01: Actividades Scripts (I)

```
/home/usuario  
/tmp  
"Contenido listado del ls"  
$pwd  
/home/usuario
```

Bash. Antonio J. León Delgado