**Práctica 05**

**El shell y su ambiente básico de trabajo.**

**INDICACIONES SOBRE EL DESARROLLO**

En todos los puntos que sigue tendrá que explicar con detalle cómo logró llevar a cabo lo pedido, indicando trayectorias, comandos con su despliegue y / o acciones realizadas. En el caso de despliegue de comandos explique el significado de lo desplegado. Cuando haya preguntas específicas, no basta con aplicar y desplegar el resultado del comando, deberá usted contestarlas. Tendrá que abrir una terminal-ventana, en Ubuntu, y también puede ayudarse del File Manager.

**PREVIO AL INICIO**

* Copie el contenido del fólder *LabAE05* de Sistemas Operativos (SO) de Comunidad. Este fólder contiene tanto el texto de la práctica como material de soporte.
* Después arranque la máquina virtual de Ubuntu y entre a la cuenta de “sisops”.

**ACTIVIDADES INICIALES**

Lea el archivo *UbuntuShell.pptx*, en formato de notas, que contiene el material de soporte de la práctica.

Desde su directorio base (*home*), copie el archivo  ***uctar*** del URL ***ftp://ftp.itam.mx/pub/investigadores/rrios/SisOpe/Lab/PracAE05/.*** Esto lo puede hacer, ya sea desde su browser, o desde su computadora Linux usando el comando ***wget***. Cuando use ***wget*** incluya en el URL, al final, el archivo ***uctar***. Una vez en el directorio base aplique el comando ***tar –xvf uctar*** que descomprimirá el archivo ***uctar*** en su directorio base creando el directorio (folder) ***uc*** que contiene a su vez varios directorios y archivos a ser usados durante esta práctica.

**DESARROLLO**

1. (pág. 4.5 del *pptx*) **A)-** Ejecute el comando **ps *-l*** anotando el árbol jerárquico de procesos (incluyendo su PID respectivo). **B)-** Cree temporalmente un subshell **sh**. Al estar dentro del proceso subshell ejecute **ps *–l*** mostrando el árbol jerárquico de procesos (incluyendo su PID respectivo). ¿De quién es hijo el subshell? Al final termine con el subshell.
2. Cree temporalmente un subshell **csh**. Al estar dentro del proceso subshell ejecute **ps *–l*.** Al final termine con el subshell. Ahora habrá regresado al shell *tcsh* para llevar a cabo el resto de la práctica.
3. (pág. 4.11)Despliegue las variables de shell.
4. Primero verifique que la variable de *shell* ***filec*** no este activada (vea que no aparezca en la lista de variables de *shell*); entonces actívela. Recuerde que ***filec*** es de tipo switch; en esta distribución de Ubuntu dicha variable viene activada. Ahora haga una prueba de funcionamiento de la terminación automática de nombres de archivos o directorios.
5. Active la variable de shell ***noclobber***, vea que aparezca en la lista de variables de shell y haga una prueba de funcionamiento que no le permita a un redirector reemplazar un archivo existente.
6. Desactive la variable de shell ***noclobber***. Liste el total de variables activas y verifique que dicha variable ya no aparezca. También verifique que ya fue desactivado haciendo la misma prueba que en el inciso anterior.
7. Elabore su propia variable de shell que contenga su propio nombre.
8. (pág. 4.16)Imprimir: **a)** el *path* del shell con una variable de shell; y **b)** el directorio base con una variable global (entorno o de ambiente).
9. (pág. 4.17)Despliegue los **alias** existentes.
10. Crear **alias** ***ldir***. Este alias al ejecutarse deberá ser equivalente al comando ***ls –l***.
11. Que hace el **alias**  ***cam ‘cd \!\*; echo $cwd’***.
12. Haga un ejemplo con el comando copiar *cp*.
13. Liste los nombres de los archivos que se encuentran en **ua**.
14. (pág. 4.18)Muestre el tamaño de la lista de eventos de **history** e incrementela en dos. Compruebe que aumentó el tamaño de la lista.
15. (pág. 4.15)Modifique la variable de shell ***prompt*** para que ahora despliegue su *nombre*, seguido del *número de evento* y los símbolos “*+>”*.
16. Busca en la lista de eventos el comando que utilizaste para la pregunta 13. Con los mecanismos de sustitución de eventos. Lista el contenido del **directorio $home/ua**.
17. Inmediatamente después de haber ejecutado los comandos de las preguntas, ejecute el siguiente comando y ayudándose del desplegado explique qué hace **!ls:0 –R1 !cp:$**
18. Después de haber ejecutado un listado simple de tu directorio (ls). Utiliza únicamente el comando de repetición!! para realizar:
19. Un listado largo de tu directorio
20. Un listado largo del directorio raíz
21. Un listado largo del directorio $home/ua
22. Con una sola sustitución. Un listado largo del directorio $home/uc
23. Muestre los tres primeros eventos de **history**.
24. (pág. 4.20)Ejecute el comando: ***(*echo *"raiz";* cd */;* pwd*);* echo *"mibase";* pwd**. ¿Quién ejecuta los comandos que están entre ( )? ¿Para qué sirve el “;”? ¿Explique que hizo todo el comando? ¿Por qué los ***pwd*** reportan resultados distintos?
25. Active la variable de shell ***notify*** y posteriormente ejecuta el comando**: *(*sleep *5;* echo “desperte”*)&***. Inmediatamente ejecute el comando **ps -l** para que ver la situación de los procesos. El carácter de **&**, al final del comando, sirve para ejecutar el proceso en el background (desligándose parcialmente de la terminal TTY). De lo desplegado por el comando **ps -l** elabore un árbol jerárquico de procesos (incluyendo su PID respectivo). ¿Hay algún proceso que le parezca fuera de lugar? ¿Esta situación la provocan los paréntesis?
26. (pág. 4.28 – 4.30) ¿Qué efecto tienen los siguientes comandos?

prompt> cat < $HOME/ua/f-rojas/Nequeteje >arch

prompt> cat $HOME/ua/f-rojas/Nequeteje >arch

prompt> cat $HOME/ua/f-rojas/Nequeteje | tee arch > /dev/null

1. Para los siguientes comandos, trata de predecir el resultado que se obtendrá. Ejecuta los siguientes comandos y asegúrate de comprender qué sucede en los casos en que no hayas obtenido los resultados esperados. Esto se debe de hacer en el directorio **uc**.

prompt> date; who | wc -1

prompt > (date; who) | wc -1

prompt > echo \*

prompt > echo ‘\*’

prompt > echo \\*

prompt > echo x\*y

prompt > echo x’\*’y

Esta práctica deberá ser entregada en COMUNIDAD. Puede ser realizada por dos o hasta por tres personas.

NO HACER

1. ***term*** es una variable del shell y ***TERM*** es una variable del medio ambiente (global). a) Asígnele a ***term*** el valor *vt200* y vea que pasa en el contenido de las dos variables. b) Ahora, asigne a ***TERM*** el valor *vt100* y vea que pasa con ambas variables. ¿Excepciones?
2. Ejecute el comando**: *(*echo *$term;* set *term=vt200;* echo *$term);* echo *"mibase";* echo *$term***. ¿Qué pasa con todo el comando? Explique.