

Universidad Tecnológica de Panamá
Sistemas Operativos I
Experiencia Práctica 4

Profesora Aris Castillo de Valencia
Estudiante: Gabriel Diaz
Cedula: 20-53-5198
Grupo: 1IF131

Objetivos:

- Probar y distinguir distintos comandos para trabajar con archivos en línea de texto, incluyendo:
 - Desplegar los procesos activos en un momento en particular y ver su estado.
 - Crear procesos nuevos.
 - Eliminar procesos

Procedimiento:

Lea cuidadosamente la guía; pruebe cada uno de los comandos listados prestando especial atención a los resultados obtenidos y a las variantes que le ofrecen las opciones de los comandos. Ponga en práctica los comandos aprendidos haciendo los ejercicios sugeridos. Llene la autoevaluación y retroalimentación y súbala a la plataforma Moodle.

Ejercicio 1:

1. Abrir la terminal de consola y escriba el comando **ps -auxf**. Describa la salida.

R/: Con el comando “ps” muestra la información de los procesos activos, los prefijos “-auxf” son filtros de la información que se necesita visualizar siendo cada uno de ellos:

- a: muestra los procesos asociados a un TTY (Terminal)
 - x: muestra los no asociados
 - u: orientado al usuario, uso del procesador, uso de memoria y su estado
 - f: muestra el formato largo de UID (usuario que lo ejecuto), PPID (ID del proceso padre)
2. Ahora abra otra terminal de texto. Nuevamente escriba el comando y redireccione a un archivo llamado procesoshoy.txt. Este archivo contendrá la salida del comando ps. Después de ejecutar el comando, revise el archivo.
ps -auxf > procesoshoy.txt



procesoshoy.txt

R/:

Ejercicio 2:

En el ambiente gráfico abra dos aplicaciones distintas. Verifique con el comando correspondiente e identifique la información sobre dicho proceso.

¿Qué ID tiene el proceso?

Comando utilizado: Top

Aplicación 1: Calculadora

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1903	alex09	20	0	1104176	51568	39016	S	29.1	2.5	0:02.18	gnome-calculato

Aplicación 2: Reproductor de música Rhythmbox

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1930	alex09	20	0	1627980	107788	59900	S	11.6	5.3	0:11.98	rhythmbox

¿Qué otra información le brinda el sistema sobre dicha aplicación?

PID: El id del proceso

USER: Usuario que está ejecutando la aplicación

%CPU: Porcentaje de uso del procesador

%MEM: Memoria RAM utilizada por la aplicación

Investigue y describa el significado de la información desplegada.

Con el comando “top” también podemos utilizar para visualizar otras características del sistema operativo desde el terminal siendo el comando “man top” una guía sobre todo lo que podemos usar para administrar desde el terminal.

Entre los comandos encontramos:

Top -o %CPU: Ordenara los procesos con los valores de uso del cpu de mayor a menor.

Top -u usuario: Mostrara los procesos del usuario que este caso colóquenos en el comando.

Top -d 5: Nos permite asignar el tiempo transcurrido en segundo entre cada muestreo que se presentara de los procesos.

Ejercicio 3:

También puede crear procesos programándolos con un archivo ejecutable, a través de la función **fork()**. Las funciones **getPID()** obtiene el ID del proceso, mientras que **getPPID()** obtiene el ID del proceso padre. Hagamos un programa en C para esto. Usaremos el programa VI para escribir el código y luego se compilará y ejecutará el programa.

Paso 1: Ingresar vi: vi nombre del archivo.c

Escriba el código del siguiente programa:

```

#include <sys/types.h>
#include <stdio.h>
main( )
{
    pid_t pid;
    int i;    int n = 10;

    for (i = 0; i < n; i++)
    {
        pid = fork( );
        if (pid != 0)
            break;
    }
    printf("El padre del proceso %d es %d\n", getpid( ), getppid( ));
}

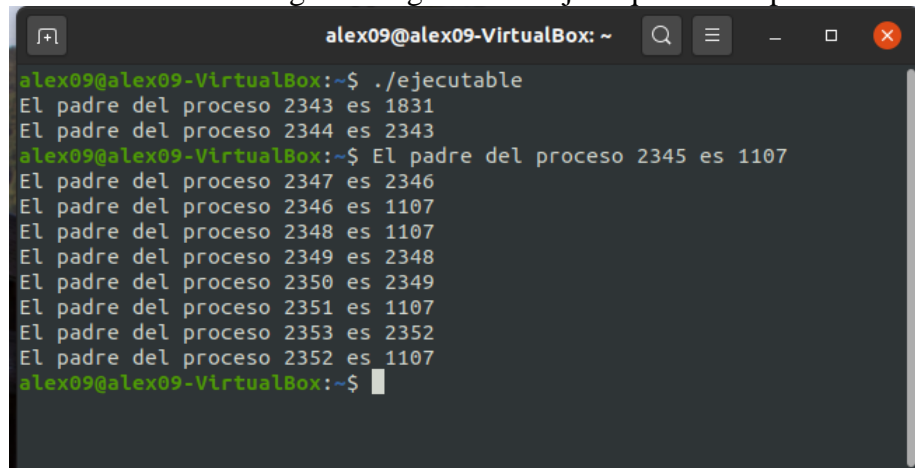
```

Para guardar el código y salir del editor vi, ejecute :wq

Paso 2: Compile el programa: `gcc -o programaejecutable programa.c`

Paso 3: Ejecutar el programa: `./programa`

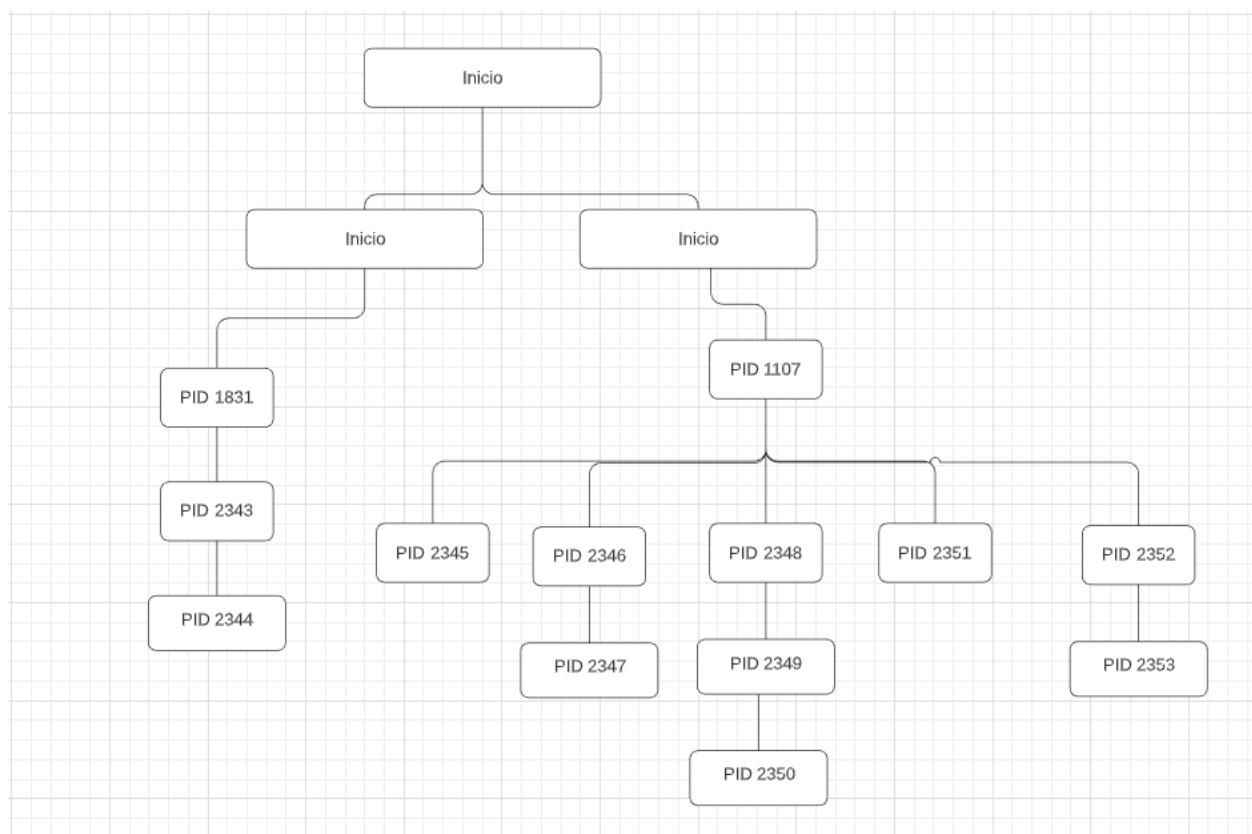
Describe la salida. Haga un diagrama de la jerarquía de los procesos creados.



```

alex09@alex09-VirtualBox: ~
alex09@alex09-VirtualBox:~$ ./ejecutable
El padre del proceso 2343 es 1831
El padre del proceso 2344 es 2343
alex09@alex09-VirtualBox:~$ El padre del proceso 2345 es 1107
El padre del proceso 2347 es 2346
El padre del proceso 2346 es 1107
El padre del proceso 2348 es 1107
El padre del proceso 2349 es 2348
El padre del proceso 2350 es 2349
El padre del proceso 2351 es 1107
El padre del proceso 2353 es 2352
El padre del proceso 2352 es 1107
alex09@alex09-VirtualBox:~$

```



Ejercicio 4:

Si ha iniciado con su usuario regular, abra otra consola de comando y allí inicie una sesión con el usuario root. Inicie algún proceso con este usuario (puede ser una aplicación cualquiera). Ejecute el comando ps y top con ambos usuarios.

¿En qué se diferencia la información de cada usuario?

La diferencia más notable es en los PID, siendo los del usuario root los procesos principales iniciando desde el PID 1 “SYSTEMD”

Retroalimentación y autoevaluación.

1. Entregue cada una de las preguntas de ejercicio.
2. Busque 5 comandos relacionados con los discutidos en esta guía. Pruébelos. Describa sus usos y escriba ejemplos específicos completos, incluyendo la sintaxis y opciones utilizadas.

- Grep: es un comando que nos servirá para encontrar texto dentro del archivo que le indiquemos. Su nombre proviene de g/re/p, un comando que sirve para algo parecido en un editor de textos de Unix/Linux. Como otros muchos comandos, grep tiene muchas opciones disponibles que añadiremos en forma de letras y cada una servirá para una tarea diferente. Al combinar estas opciones seremos capaces de realizar búsquedas complejas en uno o varios archivos.
Las opciones disponibles:

- i: no diferenciará entre mayúsculas y minúsculas.
- w: fuerza que sólo encuentre palabras concretas.
- v: selecciona las líneas que no coinciden.
- n: muestra el número de la línea con las palabras de solicitadas.
- h: elimina el prefijo del nombre del archivo Unix en la salida.
- r: busca directorios recursivamente.
- R: como -r pero sigue todos los enlaces simbólicos.
- l: muestra sólo nombres de archivos con las líneas seleccionadas.
- c: muestra sólo una cuenta por archivo de las líneas seleccionadas.
- color: muestra los patrones coincidentes en colores.

Ejemplo:

grep -i imágenes /home/pablinux/Documentos/830.desktop, donde «imágenes» sería la palabra que queremos encontrar y el resto el archivo con su ruta. Este ejemplo buscaría «imágenes» en el archivo «830.desktop» sin distinguir entre mayúsculas y minúsculas.

- Ssh: ssh usuario@host: se conecta al host como usuario (ejemplo: ssh andres@miservidor)

ssh -p puerto usuario@host: se conecta al host en el puerto especificado (sustituir puerto” por el número de puerto configurado)

ssh-copy-id usuario@host: agrega la clave para el host y usuario de ese host; sirve para activar el inicio de sesión sin contraseña con el uso de claves

- Chmod: cambia los permisos del archivo “archivo” a octal, que puede especificarse por separado para “usuario”, “grupo” y “otros”. Los valores en octal se representan a continuación:
 - 4 – lectura (r de read)
 - 2 – escritura (w, de write)
 - 1 – ejecución (x, execute)
 Explicación: Para establecer permisos, se suman los valores de arriba. Por ejemplo, para asignar al propietario del archivo (usuario) el acceso total de lectura (r), escritura (w) y ejecución (x), basta con sumar el valor octal $4 + 2 + 1 = 7$. Suponiendo que desees limitar el acceso para los miembros del “grupo”, permitiendo sólo la lectura y escritura, basta con sumar $4 + 2 = 6$. Reuniendo los dos ejemplos citados, quedaría: chmod 760 (r para el usuario, w para el grupo y 0 para otros o “rw-“)

Otros ejemplos:

chmod 777: lectura (r), escritura (w) y ejecución (x) para todos (“usuario”, “grupo” y “otros”)

chmod 755: “rwx” para el “dueño” (usuario), “rw” para el “grupo” y “otros”

Para más información, escribe en la terminal: man chmod

- Wget: es una herramienta informática creada por el Proyecto GNU. Puedes usarlo para recuperar contenido y archivos de varios servidores web. El nombre es una combinación de World Wide Web y la palabra get. Admite descargas a través de FTP, SFTP, HTTP y HTTPS.

Lista de opciones:

-o: guarda un archivo usando un nombre diferente. Ejemplo:
`wget -O wordpress-install.zip https://wordpress.org/latest.zip`
 el recurso descargado se guardará como wordpress-install.zip en lugar de su nombre original.

3. ¿En qué situaciones específicas considera que serían útiles los comandos utilizados?

Todos los comandos anteriormente utilizados en la guía son esenciales para trabajar en entornos donde no hay una interfaz gráfica de usuario (GUI), tal es el caso para distribuciones Linux para servidores donde la administración se realiza por una terminal de comandos (CLI), manejar todos estos comandos nos facilitará la administración y el correcto funcionamiento de los servidores.

4. ¿Qué dificultades encontró durante el desarrollo del laboratorio?

No se encontró una alta dificultad.

5. ¿Qué mejoraría de esta experiencia de laboratorio?

El laboratorio es muy guiado y ayuda a la investigación más profunda del tema.

Referencias:

1. Kernighan, B. y Pike, R. El Entorno de programación Unix. Prentice Hall.
2. Linux Shortcuts and Commands: <http://www.unixguide.net/linux/linuxshortcuts.shtml>
3. TTY Desmystified: <http://www.linusakesson.net/programming/tty/index.php>
4. 20 Linux System Monitoring tools every sysadmin should know: <http://www.cyberciti.biz/tips/top-linux-monitoring-tools.html>