



**Trabajo Práctico N° 4**

**Administración de CPU – Procesos**

1. Instalar compilador de lenguaje C (gcc) y librerías necesarias. Además realice una rutina en este lenguaje que sume los 5 primeros números naturales.
2. Realizar una rutina en lenguaje C que pida cargar un vector de 10 elementos numéricos y los devuelva ordenado de menor a mayor.
3. Hacer un script que realice los siguientes pasos:
  - 1- Crear un archivo de nombre process (usando comando touch).
  - 2- En el archivo process agregar todos los procesos que se están ejecutando en el sistema
  - 3- Del archivo process cortar el nombre usuario y PID y mandarlos al archivo solousu
  - 4- Comprimir y empaquetar el archivo process y solousu como proc-usu.tar.gz
4. Crear un script al cual se le pase como argumento un nombre de usuario y muestre los procesos que está ejecutando ese usuario. En caso de que no se pase ningún argumento, debe mostrar todos los procesos en ejecución, y en caso de que el usuario pasado como argumento sea root, mostrar un mensaje de error
5. Crear un scripts simple que genere un bucle infinito, luego hacer lo siguiente:
  - a) Ejecutar el mismo en segundo plano
  - b) Ejecutar el comando para ver los procesos en ejecución
  - b) Ejecutar el comando para verificar el proceso en forma online
  - c) Matar el proceso ejecutado en el punto a
6. Crear un script simple que genere una iteración de 1 a 10, por parámetro ingresar en que momento quiere cortar esa iteración y devolver el control del bash. (Sugerencias: utilizar comando "pgrep" y sleep para retrasar la ejecución y observar el resultado ).
7. Crear una rutina en lenguaje C donde se crea un proceso, como salida imprima el siguiente mensaje:

Soy el padre, si estoy en proceso padre  
Soy el hijo , si estoy en proceso hijo
8. Crear una rutina en lenguaje C donde se crea un proceso hijo, esta rutina debe tener como salida lo siguiente:

Soy el hijo, "Nro de mi ProcessID", Mi Padre es "ProcessID del Padre"  
Soy el padre, "Nro de mi ProcessID", Mi Padre es "ProcessID del Padre del Padre"
9. Crear una rutina en lenguaje C donde se crea un proceso hijo y muestre por consola los IDs del:
  - Hijo
  - Padre
  - Padre del Padre (en este caso sería el proceso del bash que llama al ejecutable)Idem: para 2 procesos hijos



**10.** Crear una rutina en lenguaje C, donde se cree un proceso hijo, además de imprimir los PID del padre e hijo, el hijo debe ejecutar el comando listar directorio y mostrar la salida del mismo por pantalla.

**11.** Crear una rutina en lenguaje C, donde se cree un proceso hijo, además de imprimir los PID del padre e hijo, el hijo debe ejecutar el comando listar directorio con el parámetro "la" y mostrar la salida del mismo por pantalla y el padre debe crear una carpeta en el directorio corriente de nombre "MisDocumentos", y un archivo de nombre "procesos" dentro de "MisDocumentos"

**12.** Crear una rutina en lenguaje C, donde me permita ejecutar varias sentencias fork(), para eso deberá ingresar por teclado la cantidad que desea ejecutar.

- a) Que muestre solamente por pantalla todos los procesos hijos
- b) Que muestre solamente por pantalla todos los procesos padres
- c) Que muestre por pantalla ambos (Padres e Hijos)

Nota: para que las salidas no sean engorrosas no ejecutar más de 4 fork()

**13.** Crear una rutina en lenguaje C, donde:

- a) El padre solamente pueda crear procesos,
- b) Que solamente un hijo puede crear procesos
- c) Que solo un hijo cree procesos en forma lineal

**14.** Hacer un script donde el mismo solicite adivinar el PID del script que estoy ejecutando, el mismo deberá informar si es mayor o menor el número que se está ingresando. Una vez adivinado mostrar por pantalla la cantidad de intentos que uso y el número de PID

**15.** Hacer una rutina en lenguaje C, que genere el siguiente árbol de procesos, donde el padre debe ejecutar un scripts de Linux (función system), el hijo 1 debe crear a su vez 3 hijos (H11, H12, H13 y mostrar el pid de cada uno), y el hijo 3 debe ejecutar un comando usando la función execl.

