Proyecto 1

Sistemas Operativos

Segundo Semestre 2021, Prof. Cecilia Hernández

Fecha Inicio: Jueves 23 de Sept. 2021.

Fecha Entrega: Jueves 14 de Octubre 2021 (23:59 hrs).

Trabajo en grupo: Integrado con 3 o 4 estudiantes. Una entrega por grupo.

Entrega: Archivo comprimido con readme.txt software e informe.

1. Objetivos
   - Introducir a los estudiantes en el manejo de procesos concurrentes en Unix, creación, ejecución y terminación usando llamadas a sistemas fork(), exec() y wait(). Además el uso de otras llamadas a sistema como signals y comunicación entre procesos usando pipes.
2. Metodología: Trabajo en grupo de 3 o 4 estudiantes.
3. Descripción

   Desarrollo de un intérprete de comandos simple en Linux (shell). La shell a implementar será similar a las disponibles actualmente en Linux. A continuación se detalla lo requerido en su implementación.

   a) Parte 1 (3.0 puntos.)
   1) La shell debe proporcionar un prompt, lo que identifica el modo de espera de comandos de la shell.
   2) Debe leer un comando desde teclado y parsear la entrada para identificar el comando y sus argumentos (debe soportar al menos 3 argumentos).
   3) Debe ejecutar el comando ingresado en un proceso concurrente, para lo cual debe usar el llamado a sistema fork() y algunas de las variantes de exec(). Los comandos a soportar son ejecutados en foreground, es decir, la shell ejecuta y espera por el término de su ejecución antes de imprimir el promtp para esperar por el siguiente comando.
   4) Si se presiona "enter" sin previo comando, la shell simplemente imprime nuevamente el prompt.
   5) Su shell debe soportar comandos que se comunican mediante pipes, es decir debe soportar comandos del tipo mishell:$ ps -auxf | sort -nr -k 4 | head -10.
   6) Su shell además debe soportar el comando exit para terminar.
   7) Debe poder continuar si es que un comando ingresado no existe, proporcionando el error correspondiente.

waitpid( ).

b) Segunda parte (3.0 puntos)

1) Su shell debe capturar la señal Control C. Para manejar esta señal su shell debe preguntarle al usuario si efectivamente quiere salir para lo cual debe esperar confirmación y hacerlo o bien continuar y esperar el siguiente comando a ejecutar.
2) Debe agregar un comando que le permita monitorear el sistema cada x segundos por z segundos y guardar la información en un archivo de log. Su nuevo comando debe tener el prototipo:

   cmdmonset nuevocmd cmdsys x z

   donde *nuevocmd* es el nombre de su nuevo comando, *cmdsys* puede ser uno de los dos comandos de administración de sistemas en linux *vmstat* o *netstat*.