Proyecto 1

Sistemas Operativos

Segundo Semestre 2024, Prof. Cecilia Hernández

Fecha Inicio: Viernes 23 de agosto, 2024.

Fecha Entrega: Lunes 9 de septiembre, 2024 (23:59 hrs).

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: Integrado con 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. Debe entregar un informe en pdf con la descripción de lo desarrollado, el cual debe incluir el link a un repositorio donde se aloja su proyecto. El repositorio debe incluir un Readme que describa como compilar y ejecutar comandos.

   I 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 un número indeterminado de 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 -aux | sort -nr -k 4 | head -20.
   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.

II Segunda parte (3.0 puntos)

1) Su shell debe implementar un comando personalizado llamado *favs*, el cual permite mantener los comandos favoritos en forma persistente. Los comandos favoritos deben almacenarse en alguna estructura de datos que contenga los comandos en memoria cuando la shell esté activa y en un archivo cuando la shell se cierra. Cada comando debe además tener asociado un número que permita identificarlo. El comando consiste en lo siguiente:
   - favs crear ruta/misfavoritos.txt: Crea archivo donde se almacenan los comandos favoritos dada la ruta y nombre de archivo. Note que la ruta puede ser cualquiera, incluyendo directorio actual.
   - Por defecto, cada vez que el usuario ejecuta un comando en su shell se agrega automáticamente si y solo si no está en la lista de favoritos. Con la excepción de los comandos asociados al manejo de favoritos.
   - favs mostrar: despliega la lista comandos existentes en la lista con su respectivo número.
   - favs eliminar num1,num2: Eliminar comandos asociados a los números entregados entre comas.
   - favs buscar cmd: Busca comandos que contengan substring cmd en la lista de favoritos y los despliega en pantalla junto con su número asociado.
   - favs borrar: Borra todos los comandos en la lista de favoritos.
   - favs num ejecutar: Ejecuta el comando, cuyo número en la lista es num.
   - favs cargar: Lee comandos de archivo de favoritos, los mantiene en memoria y los despliega en pantalla.
   - favs guardar: Guarda comandos agregados en sesión de shell actual.
2) Definir un comando personalizado que le permita definir un recordatorio, por ejemplo para ir a comprar al super o para hacer una pausa activa despues de cierta cantidad de tiempo. El comando debe ser *set recordatorio 10 "hacer pausa activa"*. Esto le dice a su shell que debe le recuerde en 10 segundos que debe hacer una pausa para descansar. Al cabo de este tiempo, su shell desplegará ese mensaje.