# Proyecto 1

## Sistemas Operativos

## Segundo Semestre 2023, Prof. Cecilia Hernández

**Fecha Inicio: Jueves 14 de Septiembre, 2023.**

**Fecha Entrega: Viernes 6 de Octubre, 2023 (23:59 hrs).**

**Trabajo en grupo: Integrado con 3 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 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 con la descripción de lo desarrollado. Además debe incluir in readme.txt que resuma como compilar y ejecutar los comandos soportados por su shell.

   *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 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 -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.

*b*) Segunda parte (3.0 puntos)

1) Su shell debe crear un comando que permita crear un demonio (daemon) que mida y registre una línea con la información requerida del sistema en el log del sistema (localizado en "/var/log/syslog") cada $t$ segundos por un tiempo total de $p$ segundos, donde $t$ y $p$ son parámetros entregados por línea de comando. Para ello debe extraer y almacenar "processes" "procs_running", "procs_blocked" en el archivo "/proc/cpuinfo" disponibles en el sistema linux. En caso necesite matar el demonio simplemente use el comando killall de linux. Para definir el demonio debe investigar sobre el contenido del archivo cpuinfo y las siguientes llamadas a sistema y usarlas para programarlo:
   - fork
   - umask
   - setsid
   - openlog
   - syslog
   - closelog