

Trabajo Práctico Nro. 3 (GRUPAL):

- **Tema:** Procesos, comunicación y sincronización
 - **Descripción:** Codificar todos los programas mencionados en el presente trabajo, teniendo especialmente en cuenta las recomendaciones sobre programación mencionadas en la introducción de este trabajo
 - **Formato de entrega:** Siguiendo el protocolo especificado anteriormente. Recomendamos realizar la entrega presencial
 - **Documentación:** Todos los programas que se entreguen deben tener un encabezado y un fin de archivo. Dentro del encabezado deben figurar el nombre del programa, el trabajo práctico al que pertenece y el número de ejercicio dentro del trabajo práctico al que corresponde, el nombre de cada uno de los integrantes detallando nombre y apellido y el número de DNI de cada uno (tenga en cuenta que para pasar la nota final del trabajo práctico será usada dicha información, y no se le asignará la nota a ningún alumno que no figure en todos los archivos con todos sus datos), también deberá indicar el número de entrega a la que corresponde (entrega, primera reentrega, segunda reentrega, etc.).
 - **Evaluación:** Luego de entregado el trabajo práctico los ayudantes procederán a evaluar los ejercicios resueltos, en caso de encontrar errores se documentará en la carátula del TP que será devuelta al grupo con la evaluación final del TP y una fecha de reentrega en caso de ser necesaria (en caso de no cumplir con dicha fecha de reentrega el trabajo práctico será desaprobado). Cada ayudante podrá determinar si un determinado grupo debe o no rendir coloquio sobre el trabajo práctico presentado.
- Las notas sobre los trabajos también estarán disponibles en el sitio de la cátedra (www.sisop.com.ar) donde además del estado de la corrección se podrá ver un detalle de las pruebas realizadas y los defectos encontrados
- Importante:** Cada ejercicio cuenta con una lista de validaciones mínimas que se realizará, esto no implica que se puedan hacer otras validaciones al momento de evaluar el trabajo presentado.
- **Fecha de entrega: 25/06/2018 o 26/06/2018**, dependiendo del día que se curse la materia.

Ejercicio 1:

Se desea generar, mediante el uso de la función fork, el siguiente escenario:

- 4 procesos hijos
- 7 procesos nietos
- 3 procesos bisnietos
- 5 procesos zombies en cualquier nivel
- 2 procesos demonios

Cada padre no podrá tener más de 2 hijos.

Cada proceso deberá mostrar por pantalla la siguiente información:

Soy el proceso con PID y pertenezco a la generación N°

Pid: Pid padre: Parentesco/Tipo: [nieto, hijo, zombie]

En el proceso principal hacer una espera hasta que se presione una tecla antes de finalizar el mismo, para poder verificar con el comando ps o pstree la jerarquía de procesos generada.

Criterios de corrección:

Control	Criticidad
Compila sin errores con el script entregado	Obligatorio
Funciona correctamente según enunciado	Obligatorio
Existe algún tipo de ayuda para la ejecución del proceso	Obligatorio

Ejercicio 2:

Se deberá generar un programa que reciba por parámetros 3 valores: un path a un directorio de “entrada”, un path a un directorio de “salida” y el nivel de paralelismo.

Una vez iniciado el programa se deberá repartir todos los archivos del directorio de entrada en forma equitativa entre los N threads que se han generado. Cada thread deberá contar la cantidad de letras totales en cada archivo, generando un nuevo archivo en el directorio de salida con el mismo nombre original y cuyo contenido sea:

Hora de inicio
Número de Thread
Cantidad de vocales
Cantidad de consonantes
Cantidad de “otros caracteres”.
Hora de finalización

Adicionalmente se deberá mostrar por pantalla:

- Cada thread, su número y qué archivos le tocó analizar
- El nombre del archivo con menor cantidad de caracteres totales.
- El nombre archivo con mayor cantidad de caracteres totales.
- El nombre del primer archivo finalizado.
- El nombre del último archivo finalizado

Criterios de corrección:

Control	Criticidad
Compila sin errores con el script entregado	Obligatorio
Funciona correctamente según enunciado	Obligatorio
Existe algún tipo de ayuda para la ejecución del proceso	Obligatorio
Valida correctamente los parámetros	Obligatorio
Acepta correctamente paths absolutos y relativos	Obligatorio

Ejercicio 3:

Escriba un proceso demonio que ofrezca un servicio de clasificación de registros recibidos a través de un FIFO, para poder guardar en distintos archivos los registros de entrada en función de la etiqueta de clasificación del mismo. Las etiquetas de clasificación serán los primeros N caracteres del registro. Cada archivo de clasificación tendrá un nombre conformado por las distintas etiquetas procesadas concatenada a la fecha de proceso ([etiqueta]_yyyymmdd).

Debe recibir por parámetro el nombre del FIFO, tamaño de la etiqueta y el directorio de destino y mostrar por pantalla el PID del proceso para poder finalizarlo luego.

No es necesario que desarrollen un proceso cliente.

Criterios de corrección:

Control	Criticidad
Compila sin errores con el script entregado	Obligatorio
Funciona correctamente según enunciado	Obligatorio
Existe algún tipo de ayuda para la ejecución del proceso	Obligatorio
Puede haber más de un proceso de envío de registros ejecutando a la vez (obviamente sobre distintos FIFOs y logs)	Obligatorio
Ejecuta como demonio	Obligatorio
Acepta correctamente paths absolutos y relativos	Obligatorio

Ejercicio 4:

Programar el juego “Preguntanos” para entorno de caracteres en GNU/Linux.

El mencionado juego será una variante el famoso juego Preguntados con la diferencia de que los distintos jugadores competirán todos al mismo tiempo por responder la pregunta que se realiza. Desde el momento en que se envía la pregunta a los participantes, se deberá dar un cierto tiempo (parametrizable) para que respondan, vencido este tiempo se pasa a la siguiente pregunta.

El juego estará compuesto por dos componentes:

1. **Servidor:** Será el encargado de realizar las siguientes acciones:

- Inicializar las preguntas y respuestas que se repartirán a los clientes: Esta acción se realizará al iniciar el proceso realizando la lectura de un archivo de texto con el siguiente formato:
P: ¿Qué tipo de interrupción es un stack overflow?
R: Software
RC: Hardware – Interna
R: Hardware – Externa
R: Ninguna de las anteriores
El archivo de parametrización de preguntas deberá tener siempre la pregunta, y todas las posibles respuestas (no debe existir un límite de preguntas y respuestas) y todas las preguntas deben tener una sola opción correcta, la cual será indicada con el prefijo RC, según se ve en el ejemplo.
- Aceptar las conexiones de los clientes (jugadores) que desean participar del juego: Esta acción se realizará luego de inicializar las preguntas y respuestas, tendrá una duración máxima que podrá estar indicada por un parámetro del binario, en un archivo de configuración o fijo en el código según lo determine el cliente.
- Establecer la comunicación con los clientes: Será el encargado de recibir y enviar los mensajes a los clientes que se hayan conectado durante el período de aceptación de clientes.
- Administración del juego: Enviará las preguntas a los clientes (todos al mismo tiempo deben recibir la pregunta) y determinará el cliente que respondió más rápido asignándole un punto al cliente si respondió bien o descontándole uno si respondió mal.
- Mantener y mostrar las estadísticas de puntos de cada jugador: En todo momento deberá mostrar en la pantalla del cliente o del servidor el resultado actualizado del juego.
- Determinar el ganador del juego: Una vez que se terminen las preguntas determinará al/los ganadores del juego.

2. **Cliente:** Será el encargado de realizar la comunicación con el jugador, realizando las siguientes acciones:

- Mantener la comunicación con el servidor: Una vez establecida la comunicación con el servidor del juego y que el servidor haya informado que fue aceptado en el juego deberá mantener activo el canal de comunicación que se utilizará para el intercambio de los mensajes
- Interfaz con el usuario: será el encargado de presentar las preguntas a los usuarios (formateo, forma de seleccionar la respuesta) y aceptar la respuesta del usuario (teclado / mouse, etc)

Criterios de corrección:

Control	Criticidad
Funciona correctamente según enunciado	Obligatorio
Maneja clientes simultáneos ilimitados	Obligatorio
Manejo adecuado ante caídas tanto del servidor como de los clientes	Obligatorio
Permite pasar tanto IP como nombre de máquina	Obligatorio
Funciona correctamente ante la muerte normal de alguno de los procesos intervinientes	Obligatorio
Utiliza sockets y threads en el servidor	Obligatorio
Maneja preguntas y respuestas ilimitadas	Obligatorio
Valida que cada pregunta tenga solo una respuesta correcta.	Obligatorio
Se adjuntan archivos de configuración (si corresponde) y archivos de preguntas/respuestas por parte del grupo.	Obligatorio
El cliente puede abandonar el juego en cualquier momento	Obligatorio

El servidor puede ser detenido en cualquier momento y declarar un ganador sin esperar la finalización de las preguntas	Obligatorio
Maneja correctamente el cierre de los puertos al finalizar	Deseable
Utiliza threads en el cliente	Deseable
Tiene archivo de configuración tanto para el cliente como el servidor (no incluye al de preguntas)	Deseable
El cliente muestra el estado de posiciones del juego en todo momento	Deseable
Tiene tiempo máximo para recibir las respuestas de los clientes, declarando que nadie ganó en caso de ocurrir esta situación	Deseable

Ejercicio 5:

Escriba un proceso que ofrezca un servicio de Encriptación/Desencriptación de datos, que debe recibir los mensajes a encriptar/desencriptar a través de memoria compartida. Si un usuario envió un mensaje a encriptar, hasta que no lo lea desencriptado, no se debe aceptar nuevos mensajes de otros usuarios. Una vez que el usuario lee el mensaje, el servidor debe eliminarlo de la memoria compartida para evitar que otro proceso usuario pueda leer el mensaje.

En la memoria compartida solo debe haber mensajes de usuario.

También generar el proceso usuario, que lea el mensaje a transformar por standard input y muestre por standard output el mensaje transformado.

El mecanismo de encriptación queda a elección del grupo, mientras que sea reversible.

Criterios de corrección:

Control	Criticidad
Compila sin errores con el script entregado	Obligatorio
Funciona correctamente según enunciado	Obligatorio
Existe algún tipo de ayuda para la ejecución del proceso	Obligatorio
Maneja correctamente la desconexión de clientes	Obligatorio
Los clientes manejan correctamente la desconexión del servidor	Deseable
No quedan recursos abiertos luego de finalizar todos los procesos	Obligatorio