Guía de Trabajos Prácticos

Esta guía contiene los Trabajos Prácticos (TP) obligatorios del curso R1042, incluyendo una guía para su entrega con el formato de presentación y un cronograma con las fechas de entrega.

Durante el transcurso del año de deberán entregarán Trabajos Prácticos, que se caracterizan por ser incrementales. Esto significa que en cada TP se irán introduciendo nuevos conocimientos con respecto al TP anterior. A su vez, se deberán utilizar las herramientas SVN, Makefile y Doxygen cuando se lo indique.

Formato de presentación

- Los archivos fuentes deben tener en todos los casos los comentarios necesarios para clarificar su lectura.
- Cada subrutina/función, debe contar con un encabezado describiendo la operación que realiza, los parámetros que espera como entrada, y los resultados que debe presentar, indicando formato y método de entrega.
- Como encabezado del programa, debe haber un comentario que explique claramente que hace dicho programa, y las instrucciones detalladas (comandos) para su compilación y "linkeo".
- TODOS los ejercicios son obligatorios.
- **La entrega de TODOS los trabajos prácticos es obligatoria para regularizar la materia.**

Trabajo Práctico N°10

- Procesos e IPC -

Ejercicio 1

Corra el siguiente programa:

```
#include <unistd.h>
#include <sys/types.h>
#include <stdio.h>
int main(void){
        pid_t pid_hijo;
        pid_hijo = fork();
        if(pid_hijo>0){
                puts("Padre: Empezando a dormir");
                sleep(20);
                puts("Padre: Fin programa");
        }
        else{
                puts("Hijo: Empezando a dormir");
                sleep(10);
                puts("Hijo: Fin programa");
        }
return 0;
}
```

En base a las pruebas realizadas y revisando el estado de los procesos en la ejecución del programa responda:

a) ¿Cómo se puede definir un proceso zombie?

b) ¿Ocupa memoria en el sistema?¿Cuánta?

c) ¿Qué sucede si se intenta matar el proceso usando el comando kill?

Ejercicio 2

Implemente un Cliente-Servidor concurrente que cumpla los siguientes requisitos:

Servidor:

Se ejercutará de manera concurrente, esperando conexiones por el port TCP 8145. Cada instancia de conexión deberá manejarse mediante un proceso separado, y deberá tener presente el uso de todos los recursos IPC vistos en clase que considere necesarios.

Por cada pedido de conexión devuelve al cliente remoto el string: “Conexión aceptada”.

Termina cuando el usuario ejecuta: CNTRL+C

Cliente:

Conecta con el servidor en el puerto indicado. Al recibir el string de “Conexión Aceptada” la presenta en pantalla y finaliza su ejecución.

Ayuda: Utilice el código de ejemplo visto en clase. Para la implementación puede utilizar: fork(), select(). No olvide comprobar el correcto funcionamiento mediente la red con sus compañeros!