Sistemas Operativos I. RECUPERATORIO Primer examen parcial. 2025.
Facultad de Informática. Universidad Nacional del Comahue

NOTA: para la resolución del examen se puede utilizar como material de referencia sus archivos de solución de los trabajos prácticos en sus cuentas Linux, o el material disponible en la web de la materia. http://se.fi.uncoma.edu.ar/so

En ningún caso se pueden utilizar herramientas online (google, chat, gpt, deepseek, etc).

Ejercicio 1.
Desarrollar un programa en Linux que realice creación de procesos y los controle. Este programa debe crear dos procesos hijos.

Cada hijo debe calcular la suma de todos los dígitos de cada número natural. Luego de calcular la suma de todos los dígitos para un número natural, debe mostrar en pantalla el resultado.
Ejemplo de salida del hijo1:
hijo 1. La suma de todos los dígitos de 1 = 1.
hijo 1. La suma de todos los dígitos de 2 = 2.
….etc
hijo 1. La suma de todos los dígitos de 13 = 4.
…etc

La suma de todos los dígitos de un número es directo. Por ejemplo, la suma de todos los dígitos del número 1234 es:
1234 ⟶ 1 + 2 + 3 + 4 = 10
Ambos hijos hacen los mismos cálculos y presentan sus resultados en pantalla.

El padre espera pulsaciones del teclado y realiza las siguientes acciones:

- Si el usuario presiona la tecla a, el padre suspende al hijo 2 y deja ejecutando al hijo 1.
- Si el usuario presiona la tecla b, el padre suspende al hijo 1 y deja ejecutando al hijo 2.
- Si el usuario presiona la tecla c, el padre deja ejecutando ambos hijos.
- Si el usuario presiona la tecla q, el padre finaliza los hijos y termina su ejecución.

Para suspender un proceso en UNIX/Linux se puede utilizar el siguiente system call

```
#include <signal.h>
kill(pid_N, SIGSTOP);  // pausar el proceso con pid pid_N
kill(pid_N, SIGCONT);  // activar el proceso con pid pid_N
```

Finalizar procesos: un proceso en UNIX/Linux puede finalizar otro proceso realizando el siguiente system call:

```
#include <signal.h>
kill(pid_N SIGKILL);        // finalizar el proceso con pid pid_N
```

Mas información con

```
man 2 kill          # para leer la página de manual de la función de C kill() que realiza el system call
```

Sistemas Operativos I. RECUPERATORIO Primer examen parcial. 2025.
Facultad de Informática. Universidad Nacional del Comahue

Ejercicio 2.
Agregue a XINU su implementación de ahorcado para XINU, con la siguiente modificación:

- El proceso padre creará un proceso hijo, quien se encargará de la lógica de todo el juego.
- El proceso padre debe quedar verificando algún elemento del juego que le indique el estado del juego: si el jugador ganó, perdió o aún está jugando.
- SI el jugador está aún "jugando", el padre delega el procesador por un segundo. Luego de ese tiempo, el proceso padre vuelve a verificar el estado del juego.
- Cuando el proceso padre detecta que el jugador perdió o ganó, finaliza al proceso hijo y emite un mensaje al usuario informando que perdió (o ganó).

IMPORTANTE 1: Para la resolución utilice UNICAMENTE el siguiente codigo fuente de XINU:
https://se.fi.uncoma.edu.ar/so/misc/xinu-pc.tar.gz
(SI REUTILIZA el código fuente de XINU que usó durante los trabajos prácticos SE ANULA LA ENTREGA).

IMPORTANTE 2: Debe utilizar la resolución del ahorcado QUE ENTREGÓ o una versión posterior funciona. Pero NO OTRA solución (es decir, la que entregó en pedco o una posterior funcional, Corregida en caso de que su entrega no haya estado funcional).

IMPORTANTE 3: Se entrega todo el código fuente de XINU, no sólo el fuente .c . Antes de crear un zip o tar.gz de la carpeta xinu-pc realice un make clean en compile/

Ejercicio 3. Responda en un archivo .txt

a. Describa cómo se ejecutan los siguientes procesos en un sistema operativo que utiliza un planificador de CPU Shortest-Job-First Scheduling preemptive.

Proceso	Prioridad	Ráfaga(ms)	Arribo a la cola de listos
P1	3	15	35
P2	3	20	0
P3	4	20	20
P4	2	20	12
P5	4	5	20
P6	1	15	8

b. COPIE y PEGUE su resolución del ejercicio 1.c., del trabajo práctico 2.

c. COPIE y PEGUE su resolución del ejercicio 10.a., del trabajo práctico 0.