**Sistemas Operativos I. Recuperatorio del Segundo 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.** Se requiere desarrollar un programa en LINUX que, de un archivo mezcla.txt que contiene el contenido de dos libros, separe los mismos y genere dos archivos nuevos con los libros separados.
Se entrega con este ejercicio el archivo grande llamado *mezcla.txt*, que contiene los textos de dos libros mezclados. Puede obtenerlo en este link

https://se.fi.uncoma.edu.ar/so/misc/mezcla.txt

Este archivo tiene la siguiente estructura:

- La primera oración pertenece al primer libro.
- La segunda oración pertenece al segundo libro.
- La tercera oración pertenece al primer libro.
- La cuarta oración pertenece al segundo libro.

Es decir, las oraciones impares pertenecen al primer libro y las pares al segundo.

Las oraciones terminan con punto.

a. Crear una función que devuelva la cantidad de bytes que contiene el archivo *mezcla.txt* utilizando la técnica de recorrer el archivo hasta el final y contando los bytes.
b. Desarrollar un programa que utilice la función en a. , y solicite al sistema operativo memoria dinámicamente para contener los dos libros ya separados y el archivo original (si *mezcla.txt* es de N bytes, puede pedir N bytes para cada libro, de modo de estar seguros que quepa cada uno). Puede llamar a cada memoria (nombres de los punteros): `original`, `libroA`, `libroB`.
c. Cargar en la memoria llamada "`original`" el archivo *mezcla.txt*
   Luego, el programa debe colocar la primera oración del libro original (ya en memoria) en la memoria dinámica `libroA` solicitada. La segunda oración de original en la memoria dinámica `libroB` solicitada. Y así sucesivamente ir colocando las oraciones impares en `libroA` y las pares en `libroB`.
   Si los puntero que contienen la dirección de las memorias dinámicas son declarados de tipo char *, ejemplo:

   ```
   char * libroA;
   ```

   entonces luego puede acceder a cada letra del libro con `libroA[i]`. Es decir, simplemente se accede como un arreglo.
d. Guardar ambos libros separados (`libroA` y `libroB`) en los archivos *libroA.txt* y *libroB.txt*
   SUGERENCIA: borre (CON CUIDADO) los archivos libroA.txt y libroB.txt que va generando entre ejecución y ejecución (así se asegura que se generan de cero en cada prueba o ejecución).

**Sistemas Operativos I. Recuperatorio del Segundo examen parcial. 2025.**
**Facultad de Informática. Universidad Nacional del Comahue**

**Ejercicio 2.** Portar a XINU el programa fusiona imágenes del parcial.

Pero, de la siguiente manera:
Dos procesos, "`impares`" y "`pares`" fusionarán las imágenes de la siguiente manera:

- El proceso "`impares`" colocará cada pixel IMPAR de la imagen 1 (de `pixels1[]`) en las posiciones impares de la imagen fusionada.
- El proceso "`pares`" colocará cada pixel PAR de la imagen 2 (de `pixels2[]`) en las posiciones pares de la imagen fusionada.
- Como la imagen fusionada es un recurso compartido entre ambos procesos, utilice su mutex desarrollado para lograr un uso exclusivo del recurso cada vez que se quiera colocar un píxel en la imagen fusionada.
- Cuando cada proceso termina de colocar todos sus píxeles envía al programa principal el valor 16532.
- No es necesario colocar la cabecera en la imagen fusionada.

La diferencia principal con el ejercicio del parcial es que no es necesario solicitar, con un system call, memoria compartida. **Indique por qué con un comentario al principio del código fuente de su resolución.**

Los arreglos con las imágenes están aquí (coloque los archivos dentro del directorio shell/ de XINU):
https://se.fi.uncoma.edu.ar/so/misc/ej2-r2.tar.gz

Las imágenes ya están cargadas en los arreglos
`pixels1[]` y
`pixels2[]`

Cada imagen tiene 270000 bytes/pixeles.
No es necesario colocar la cabecera.
El XINU a utilizar es este: https://se.fi.uncoma.edu.ar/so/misc/xinu-pc.tar.gz

El programa principal, una vez creado los procesos `pares` e `impares`, se quedará esperando los dos mensajes de ambos procesos, para saber que finalizaron de fusionar. Una vez recibidos el programa principal muestra en pantalla VGA la imagen fusionada , con la función mostrar() que se presenta en la siguiente página.

IMPORTANTE: Para no perder mensajes, es apropiado darle, a los programas impares y pares menos prioridad que el programa principal (el programa principal tiene prioridad 20).
**Explique por qué haciendo simplemente eso no se pierden mensajes.**

```
extern void pixel(unsigned x, unsigned y, uint32 color);

void mostrar(unsigned char * imagen)
{
        int *c;
        int n = 0;
        for (int y= 0; y<300; y++)
        for (int x= 0; x<300; x++) {
                c = (int *) &imagen[n];
                pixel(x, y, *c);
                n = n + 3;
        }
}


// Si la imagen fusionada queda en un arreglo llamado fusionada
// se debe mostrar la imagen simplemente con :

mostrar(fusionada);
```