

Temario

- Lenguaje C. Sistema Operativo Xinu. Sistema Operativo Linux.
- Comunicación entre procesos.
- Drivers de E/S
- Administración de memoria
- Sistema de archivos

Software y Hardware

El parcial se realiza sobre las computadoras de laboratorio.

Utilice como referencia una hora para cada ejercicio.

Ejercicio 1. Implementación del driver del teclado utilizando pasaje de mensajes.

- a. Modifique su driver de teclado para que en vez de utilizar un buffer, se utilice el mecanismo de comunicación de pasaje de mensajes. Esta implementación no utiliza buffer. Simplemente, la sección upper-half, cuando se le solicita hacer una lectura de entrada, bloqueará esperando el mensaje. Cuando suceda un evento del hw del teclado, la sección lower-half del driver enviará con un mensaje la tecla presionada. Cuando el mensaje arriba a la sección upper-half, significa que una tecla ha sido presionada, y la llamada al sistema bloqueada puede continuar y retornar a su llamador.

Para este ejercicio utilice el código original de Xinu:

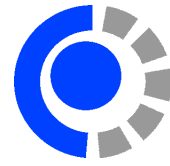
<http://se.fi.uncoma.edu.ar/so/misc/xinu-pc.tar.gz>

y reemplace los archivos del driver del teclado por su driver del teclado del TP de E/S.

- b. Verifique que funcione normalmente: Implemente un programa compuesto por dos procesos.
- Un proceso lee una entrada desde el teclado ps/2 y el segundo proceso lo muestra por pantalla.
 - El proceso que lee la entrada debe entregar la lectura del teclado vía pasaje de mensajes al segundo proceso, luego, volver a esperar una entrada.
 - El segundo proceso puede mostrar en la pantalla VGA (amarilla), utilizando un código similar a este:

```
/* Porción de pseudocódigo del proceso dos para mostrar en vga: */  
int scancode;  
  
scancode =      /* COMPLETAR esperando un mensaje del proceso 1 */  
sprintf(t, "kbd: 0x%x", scancode);  
print_text_on_vga(10, 300, t);
```

Otra opción es que el proceso 2, una vez que recibe el , presente con printf() sobre la tty (donde se ejecuta el shell).



Ejercicio 2. Implementación de un programa que contenga en memoria un libro.

- a. Desarrollar un programa en lenguaje C en Linux, que lea el contenido del libro `principe_y_mendoza.txt`, y lo coloque en una sección de memoria del programa que previamente se solicitó dinámicamente al sistema operativo. El tamaño del archivo no se conoce a priori, por lo que el programa debe obtenerlo previo a solicitar la memoria.

Descargar el archivo `principe_y_mendoza.txt` usando este comando:

wget http://se.fi.uncoma.edu.ar/so/parciales/2do_parcial/principe_y_mendoza.txt

- b. Demuestre que en la posición nro **65512** del archivo se encuentre la palabra “Catalina”.
Una solución posible es alcanzar esa posición en el segmento de memoria con el contenido del archivo, o alcanzar esa posición mencionada en el archivo abierto (usando `lseek()` u otro). Una vez alcanzada esa posición mostrar los siguientes 8 bytes en pantalla.
- c. Presente las direcciones de memoria virtual de los segmentos del programa. Responda: ¿Cuáles son las direcciones del segmento que el sistema operativo le reservó a su programa dinámicamente cuando se le solicitó?. Si necesita hacer una pausa cuando el programa está en ejecución puede agregar un bucle infinito temporalmente al final del código.

Ejercicio 3. Implementación de sistemas de archivos.

Se cuenta con un pendrive de 32GB. En una partición se ha creado un sistema de archivos FAT16. Los bloques del disco (llamados clusters en FAT) que utiliza el sistema operativo son de 4KB de tamaño.

NOTA: FAT16 utiliza 2 bytes para señalar/apuntar bloques/clusters.

- ¿Cuánto espacio del pendrive puede aprovechar esta FAT16?
- ¿Cuánto ocupa la tabla FAT en memoria para este sistema de archivos?
- ¿Cuál es el tamaño máximo de un archivo en este contexto?