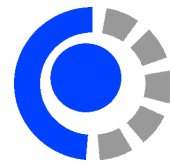




## Sistemas Operativos I 2023

### Primer parcial



#### Temario

- Lenguaje C. Sistema Operativo Xinu.
- Procesos. Creación y finalización de procesos.
- Estados de un proceso. Planificación de procesos.
- Herramientas de sincronización de procesos.

#### Software y Hardware

El parcial se realiza sobre las computadoras de laboratorio. [Utilizar la versión de Xinu que está en la web de la materia para resolver este parcial.](#) Sólo se puede utilizar la web de la materia, PEDCO, editor pluma y las herramientas de desarrollo.

En <http://se.fi.uncoma.edu.ar/so/parciales/carrera.c> se encuentra el código fuente de tres programas que harán una mini carrera: `corredor_a()`, `corredor_b()` y `muestra_carrera()`. `corredor_a` y `corredor_b` incrementan una variable global distinta cada uno, y `muestra_carrera` las muestra por pantalla.

Realice los siguientes ejercicios:

1. Implemente un proceso llamado “control\_race” que controla los procesos realizando esa carrera. Para eso, control\_race debe crear y poner a ejecutar los tres programas en `carrera.c`.
2. Agregue el programa control al shell de Xinu. Todos los procesos tienen prioridad 20. Utilice como tamaño de pila mínimo para cada proceso 1024 bytes.
3. Luego, el proceso control\_race debe permitirle al usuario gestionar los procesos que corren de la siguiente manera:
  - Si el usuario presiona la tecla 1, el proceso `corredor_a` se suspende/reanuda.
  - Si el usuario presiona la tecla 5 el proceso `corredor_a` muere.
  - Si el usuario presiona la tecla 2 el proceso `corredor_b` se suspende/reanuda.
  - Si el usuario presiona la tecla 6 el proceso `corredor_b` muere.Implemente las pulsaciones del teclado (teclas 1, 2, 5, y 6). A partir de las pulsaciones “control\_race” realiza las acciones mencionadas (suspender/reanudar, matar).
4. Modificar el programa `corredor_a()` y `corredor_b()` para que incrementen una variable entera global COMPARTIDA llamada “distancia”. Modificar control\_race para que si el usuario presiona la tecla “p” imprima el valor de la variable distancia (utilice la misma técnica que los programas corredor para mostrar esta variable en una misma línea de la consola). Nota: Si hay condiciones de carrera entre los procesos proteger esas secciones de **manera apropiada**. No proteger secciones que NO SEAN condiciones de carrera.
5. Responder:
  - a. ¿A qué estado (de los estados de Xinu) pasa el proceso “control\_race”, si `corredor_b()` le cambia la prioridad (a control\_race) para que tenga prioridad 10?.
  - b. ¿Cuando un proceso espera por un semáforo (en Xinu) y el semáforo no está disponible, a qué estado de Xinu pasa a estar ese proceso?.
  - c. ¿Cuál es el quantum de Xinu y donde está definido?. Conociendo el quantum de Xinu indique: ¿cuántas veces por segundo Xinu le asigna ráfagas de CPU a cada proceso?. ¿y por qué?.

Nota 2: entregar en la tarea de PEDCO el código fuente de Xinu con la resolución de este parcial, en un archivo llamado **nombre.apellido.legajo.tar.gz**. Entregar en un archivo txt las respuestas a las preguntas del ejercicio 5.