



UNIVERSIDAD TÉCNICA  
FEDERICO SANTA MARÍA

**Teoría de Sistemas Operativos  
ELO321  
Primer Semestre 2024**



## **Tarea 1**

### **Creación y Comunicación de Procesos Teoría de Sistemas Operativos – ELO321**

**Departamento de Electrónica  
Universidad Técnica Federico Santa María**

## 1. Objetivos

- Interiorizar al estudiante con la programación, creación y comunicación de procesos.
- Creación, escritura y lectura de Memoria Compartida.

## 2. Herramientas a Utilizar

- Sistema Operativo Linux.
- Compilador gcc.
- Lenguaje de Programación C.
- Creación de procesos, creación, escritura y lectura de Memoria Compartida POSIX en C.

## 3. Evaluación

- 3.1. La tarea debe realizarse en grupos de 3 personas máximo (puede ser individual).
- 3.2. Se evaluará:
  - Código Fuente.
  - Funcionamiento.
  - Documentación (realizada como comentarios en el código fuente).

## 4. Creación y Comunicación de Procesos

### JUEGO DE LA VIDA DE CONWAY

Se solicita generar un proceso padre, en el lenguaje de programación C, que realice las siguientes acciones:

- El proceso padre debe aceptar un número entero de entrada N (parámetro de entrada iteraciones), verificar el ingreso del parámetro y si no se entrega parámetro mostrar una ayuda de la operación del programa.
- El Proceso padre deberá crear una matriz de caracteres de tamaño “ROWS” filas y “COLUMNS” columnas, constantes definidas en el programa (ROWS=40 y COLUMNS=120) inicializadas con el carácter '0' o el carácter '1' en forma aleatoria en cada celda. Esta matriz debe ser mantenida en una memoria compartida que almacena solo caracteres (char). Debe organizar el almacenamiento de acuerdo a la matriz como un tablero de dos dimensiones.

- Creará dos procesos hijos que deberán:

- HIJO 1:

- Considerando la memoria compartida como un tablero de dos dimensiones con “ROWS” filas y “COLUMNS” columnas para implementar el “Juego de la Vida”.
- Cada casilla de este tablero se considera una celda. Cada celda tiene 8 celdas "vecinas" (salvo las celdas ubicadas en los bordes del tablero), que son las que están alrededor de ella, ejemplo: figura 1 es un tablero de 5 filas y 5 columnas con una celda en negro que la podemos considerar como “viva”.

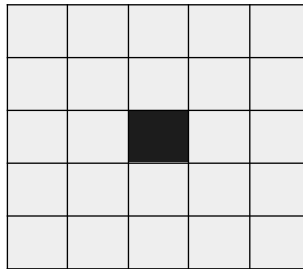


Figura 1

- Las celdas tienen dos estados: están "vivas" o "muertas" (o "1" y "0", negro o blanco, etc.). El estado de las celdas se actualizan en cada iteración de acuerdo a las siguientes reglas:
    - Nace: Si una celda muerta tiene exactamente 3 celdas vecinas vivas "nace" (es decir, al turno siguiente estará viva).
    - Muere: una celda viva puede morir por uno de 2 casos:
      - Sobre población: si tiene más de tres vecinos alrededor.
      - Aislamiento: si tiene solo un vecino alrededor o ninguno.
    - Vive: una celda se mantiene viva si tiene 2 o 3 vecinos a su alrededor.
  - Debe recorrer todas las celdas (memoria compartida) actualizando su estado, la cantidad de iteraciones está dado por la entrada N del programa.
- HIJO 2:
- Debe acceder a la memoria compartida para leer el tablero definido y mostrarlo por consola con un intervalo de tiempo razonable para que se puedan visualizar los cambios de las celdas generados por el Hijo1 al aplicar el algoritmo. Puede considerar el mismo retardo para ambos procesos hijos.
  - Utilizar la librería “Ncurses”, que es una API que permite desplegar interfaces basadas en texto que permite, por ejemplo posicionar caracteres en pantalla, asignar colores, etc. Se deja disponible un ejemplo que maneja una ventana de 40 filas por 120 columnas.
  - Debe recorrer todas las celdas (memoria compartida) mostrando su estado en

pantalla utilizando Ncurses, la cantidad de iteraciones está dado por la entrada N del programa.

- El proceso padre deberá esperar a que cada proceso hijo termine.
- Toda comunicación entre el padre e hijos se debe realizar utilizando memoria compartida.
- Debe tomar las medidas necesarias para no generar procesos huérfanos ni zombis y terminado el proceso padre liberar la memoria compartida utilizada.

## 5. Entrega y Pruebas del Código

La entrega de los códigos se realizará a través de aula.usm.cl (nombre archivo APELLIDO1\_APELLIDO2\_APELLIDO3.zip) y estos deben estar disponibles en el servidor aragorn.elo.utfm.cl. Las pruebas para el correcto funcionamiento se realizarán en este mismo servidor. Pruebe el correcto funcionamiento en el servidor aragorn antes de realizar a entrega.

Fecha de Entrega: 8 de Mayo 2024. **Se descontará 5 puntos por día de retraso.**