

Universidad de Costa Rica
Facultad de Ingeniería
Escuela de Ciencias de la Computación e Informática

Sistemas Operativos
Grupo 03
I Semestre 2023

Tarea Programada 1

Profesor:
Rafael Francisco-Arroyo

Estudiante:
Daniel Pérez-Morera | Carné: C15906
12/04/2023

Índice

1. Introducción	3
2. Objetivos	3
3. Descripción	4
4. Diseño	4
PThreads	4
ShMemory & Semaphore	4
MailBox	4
5. Manual de usuario	5
Requerimientos de Software	5
Compilación	5
Especificación de las funciones del programa	5
6. Casos de Prueba	6

1. Introducción

Este documento pretende permitir a cualquier persona poder comprender el funcionamiento de los programas adjuntos y como poder utilizar estos recursos. Este proyecto busca simular el juego de la papa caliente con varios procesos computacionales.

2. Objetivos

- Simular el juego de la papa caliente con PThreads(UNIX).
- Simular el juego de la papa caliente con ShMemory & Semaphore(UNIX).
- Simular el juego de la papa caliente con MailBox(UNIX).

3. Descripción

Esta simulación debe construir un anillo de procesos(n) simulando la ronda en el juego de la papa caliente. Donde el proceso i debe pasar la papa al proceso $i+1$ o $i-1$, menos los procesos n , quien debe pasar la papa al proceso 0 o $i-1$, y el proceso 0, quien debe pasar la papa al proceso 1 o n . La papa tiene un valor el cual representa la vida y el usuario puede modificar este. Y el estallido de la papa es represento mediante las reglas de Collatz.

Ademas como se indica en los objetivos, se desarrollaron 3 tipos de simuladores con las diferentes tecnologías que brinda UNIX, sin embargo la esencia en esto es la misma.

4. Diseño

PThreads

Este simulador fue realizado con la tecnologías de PThreads de (UNIX), donde se crean hilos y mutex para evitar las condiciones de carrera y producir la simulación de la papa. Para este simulador se tiene una clase PThread quien es el responsable de encapsular las funciones de PThreads y un archivo Simulación el cual contiene el main, lugar donde inicia los procesos, y la simulación de los participantes.

ShMemory & Semaphore

Esta programa fue creado con la funciones de ShMemory, nos brinda un área de memoria compartida la cual podemos compartir entre distintos procesos, `fork()`, permite crear varios procesos hijos del proceso principal(padre), y Semaphore, ayuda a manejar la concurrencia, condiciones de carrera y a simular el juego de la papa. Para este simulador tenemos las clases Semaphore, la cual encapsula las funciones de los semáforos de UNIX, y ShM, la cual encapsula las funciones para la memoria compartida. Ademas se tiene un archivo papa el cual contiene la función principal(main) y la simulación de la papa para los procesos creados.

MailBox

Esta simulación se realizo con la un MailBox, el cual nos brinda una cola de datos compartida entre los procesos y permite que los paquetes tengan un destinatario especifico, y `fork()`, para crear los procesos que simulan los jugadores de la papa, ambas funciones de UNIX. Para esta simulación se cuenta con una clase MailBox, la cual encapsula todas las funciones utilizadas de la cola compartida. Ademas se tiene un archivo papa el cual contiene la función principal(main) y la simulación de la papa para los procesos creados.

5. Manual de usuario

Requerimientos de Software

- Sistema Operativo: Linux
- Arquitectura:UML
- Ambiente: C++

Compilación

Se debe dirigir desde la terminal a una de las carpetas que contiene el programa a ejecutar.
Se debe ejecutar el comando:

```
$ make $
```

Ejecutar el programa con el comando:

```
$/bin/<ejecutable> <vida> <players> <rotacion> $
```

La variable ejecutable debe estar rellena con el nombre del ejecutable creado en la compilación.

La variable vida indica la vida inicial de la papa.

La variable players indica cuantos jugadores estarán en el juego.

La Variable rotación indica la dirección de rotación de la papa, donde l significa left(izquierda) y r significa Right(derecha).

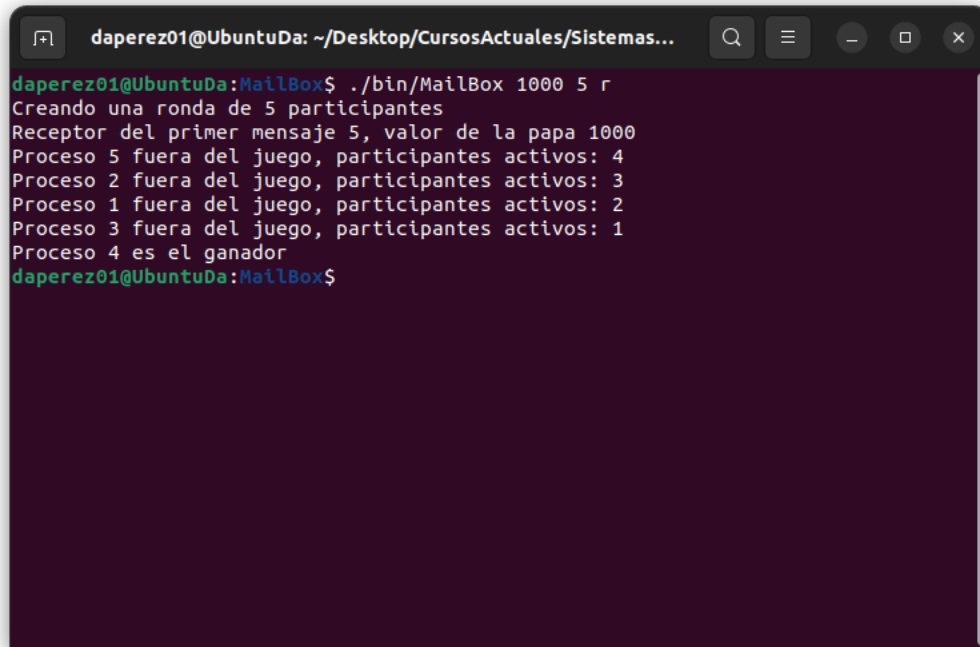
Ademas los parámetros vida, players y rotación deben ir juntos, de lo contrario el programa los ignorara y ejecutara valores predeterminados por el programa.

Especificación de las funciones del programa

Se espera que este programa muestre en pantalla el desarrollo de la papa caliente entre los distintos procesos. Esta programado para visualizar quien inicia, el orden en que los procesos salen del juego y quien gana el juego.

6. Casos de Prueba

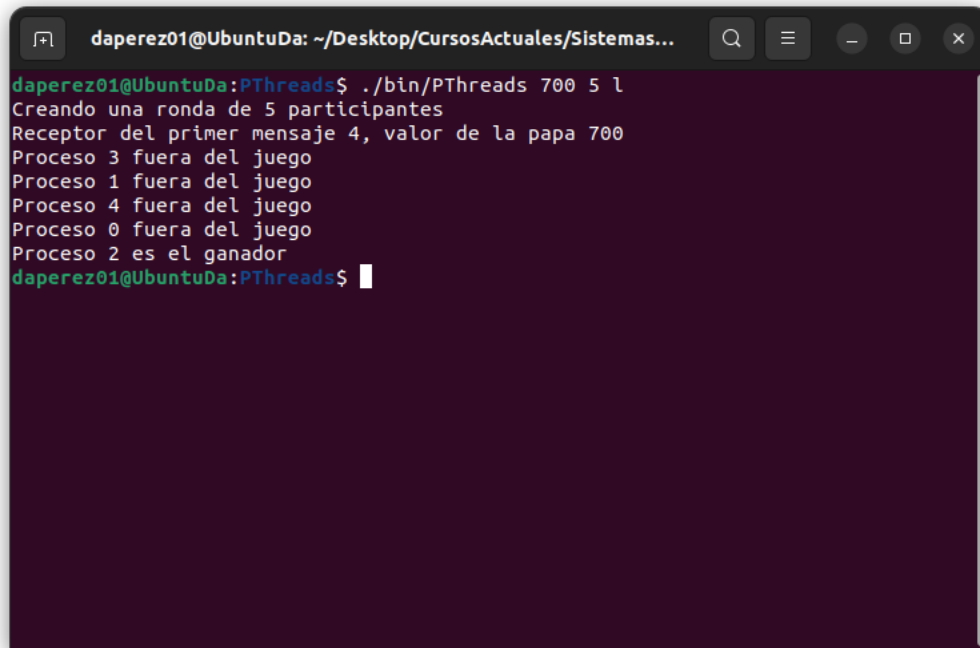
Prueba 1:

A terminal window titled 'daperez01@UbuntuDa: ~/Desktop/CursosActuales/Sistemas...' with standard window controls. The terminal shows the execution of a script './bin/MailBox 1000 5 r'. The output indicates the creation of a round with 5 participants, the receipt of a message with a value of 1000, and the elimination of participants 5, 2, 1, and 3 in sequence, leaving participant 4 as the winner. The prompt returns to the user.

```
daperez01@UbuntuDa:MailBox$ ./bin/MailBox 1000 5 r
Creando una ronda de 5 participantes
Receptor del primer mensaje 5, valor de la papa 1000
Proceso 5 fuera del juego, participantes activos: 4
Proceso 2 fuera del juego, participantes activos: 3
Proceso 1 fuera del juego, participantes activos: 2
Proceso 3 fuera del juego, participantes activos: 1
Proceso 4 es el ganador
daperez01@UbuntuDa:MailBox$
```

Figura 1: Prueba de simulación en MailBox

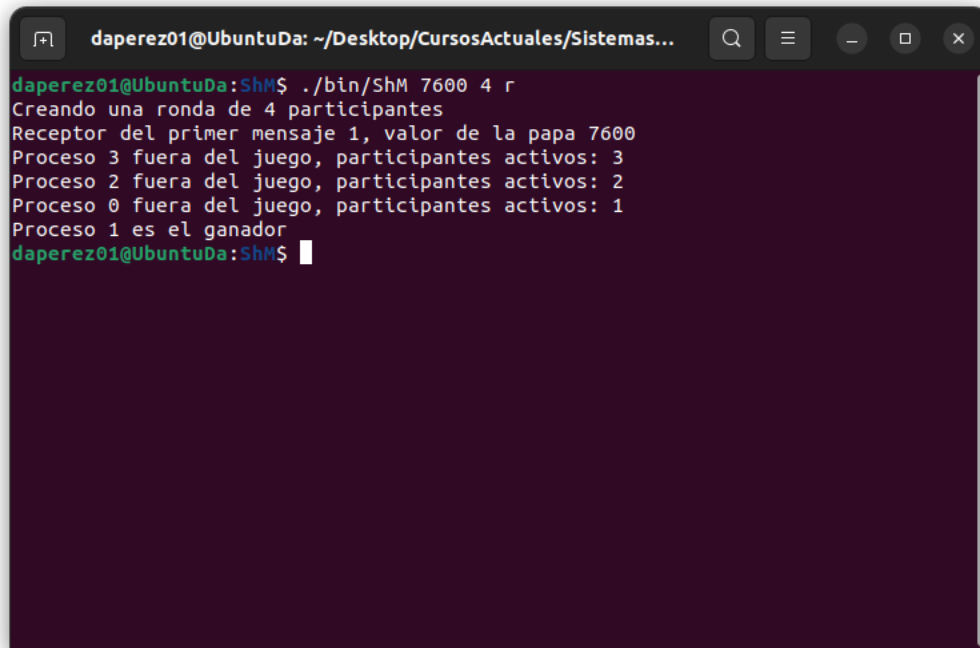
Prueba 2:



```
daperez01@UbuntuDa: ~/Desktop/CursosActuales/Sistemas...
daperez01@UbuntuDa:PThreads$ ./bin/PThreads 700 5 1
Creando una ronda de 5 participantes
Receptor del primer mensaje 4, valor de la papa 700
Proceso 3 fuera del juego
Proceso 1 fuera del juego
Proceso 4 fuera del juego
Proceso 0 fuera del juego
Proceso 2 es el ganador
daperez01@UbuntuDa:PThreads$
```

Figura 2: Prueba de simulación en PThreads

Prueba 3:

A terminal window titled 'daperez01@UbuntuDa: ~/Desktop/CursosActuales/Sistemas...' with standard Ubuntu window controls. The terminal shows the execution of a program with the command './bin/ShM 7600 4 r'. The output is in Spanish and describes a simulation of 4 participants. It shows the initial state, the removal of participants 3, 2, and 0 from the game, and finally declares 'Proceso 1 es el ganador' (Process 1 is the winner). The prompt returns to 'daperez01@UbuntuDa:ShM\$'.

```
daperez01@UbuntuDa:ShM$ ./bin/ShM 7600 4 r
Creando una ronda de 4 participantes
Receptor del primer mensaje 1, valor de la papa 7600
Proceso 3 fuera del juego, participantes activos: 3
Proceso 2 fuera del juego, participantes activos: 2
Proceso 0 fuera del juego, participantes activos: 1
Proceso 1 es el ganador
daperez01@UbuntuDa:ShM$
```

Figura 3: Prueba de simulación en ShMemory & Semaphore