



18 DE ABRIL DE 2023

PRACTICA 2

CONTROL DE LLAMADAS TELEFÓNICAS (OBELIX)

ALFONSO SEBASTIAN SIMORTE RAMOS,
KEVIN ALFONSO GÓMEZ SANDOVAL



Solución planteada

La solución planteada en nuestro caso conlleva a la utilización de 4 clases las cuales son:

- Manager
- Línea
- Teléfono
- Memorial
- Semaforol

Además, utilizaremos las siguientes interfaces

- Definitions.h
- Memorial.h
- Semaforol.h

En la clase manager se realiza la creación, lanzamiento y eliminación de los procesos tanto para las líneas como para los teléfonos, además esta clase se realizan la creación de los semáforos y la variable compartida que utilizaremos en nuestro programa y manejo de excepciones que se puedan producir durante la ejecución del programa, al terminar elimina (mata) los procesos y libera los recursos, por ultimo termina la ejecución del programa.

En la clase línea, tenemos un main el cual llama al método línea donde se obtendremos el PID, los semáforos y la variable compartida que fue creada en el proceso manager, lo primero que realiza es una espera de entre 1 y 60 segundos de manera aleatoria a continuación comprueba que nadie este modificando la variable compartida en caso de ser así, incrementa su valor en 1, posteriormente notifica que ha dejado de modificar, muestra el numero de llamadas en espera en ese momento y se realiza el juego de semáforo correspondiente.

En la clase teléfono , tenemos un main el cual llama al método teléfono donde se obtendremos el PID, los semáforos y la variable compartida que fue creada en el proceso manager posteriormente entramos en un bucle infinito donde notifica a las líneas que está disponible y espera a que ellas le necesiten a continuación comprueba que nadie este modificando la variable compartida en caso de ser así, decrementa su valor en 1, posteriormente simula una llamada cuya duración será entre 1 y 20 segundos.

Ejecutar y compilar

La realización del compilado y ejecución es sencillo, necesitamos abrir un terminal y darnos cuenta que estemos en el directorio donde se encuentra nuestro proyecto, en el caso de no ser así debemos redirigirnos a ese punto utilizando el comando `cd` e introduciendo los directorios correspondientes, cuando nos encontremos en tal punto, para compilar utilizaremos el comando `make clean`, el cual este liberara todo lo que se encuentre en la memoria, posteriormente procedemos y utilizamos el comando `make` y este nos lo compilara y posteriormente pulsamos `f5` para ver el funcionamiento del programa de esta manera los parámetros de teléfonos y líneas los tomamos del fichero `launch.json`, otra manera de poder ejecutarlo es que nosotros le pasemos por parámetros el número de teléfonos y líneas por medio del siguiente comando (situados en esqueleto, el cuales nuestra raíz) `“./exec/manager 3 10”`.

Desafíos encontrados

Algunos desafíos que encontramos durante la realización de este proyecto es que en resumen todo hacia lo que le daba la gana, con esto nos referimos que los semáforos no controlaban las líneas y los teléfonos, otro desafío fue que no terminaba los procesos y estos trabajaban en segundo plano provocando que el programa no se ejecutará correctamente.