

Programación. Tema 2 Bloque II

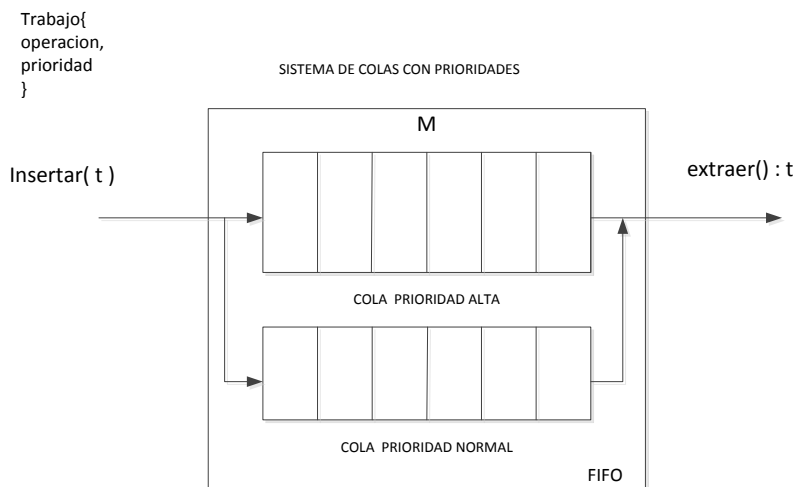
Prácticas obligatorias

Normativa:

- Las siguientes prácticas **son obligatorias** para aprobar la asignatura y tienen que **ser entregadas y explicadas antes de la fecha fijada** por el profesor.
 - El material a entregar será un **diagrama UML** y el **código fuente de la práctica en el lenguaje requerido**.
 - **No se corregirán prácticas que no compilen.**
 - Las prácticas **se realizará de forma individual**.
-

Práctica 1. C++

Se desea diseñar un **Sistema de colas con prioridades**. Para ello considere el siguiente diagrama.



Donde

- **M** es la **capacidad máxima de cada cola** (Cola prioridad alta (CPA) y Cola prioridad normal (CPN)). Tome $M=5$ para las prácticas
- **FIFO (First in, First out)** es la **disciplina** que usa el sistema de gestión de la cola: primero en entrar, primero en salir
- **Trabajo** representa un trabajo con el nombre de una operación (Ej" OP1")y una prioridad asociada (ALTA o NORMAL)

Las **operaciones permitidas** sobre el sistema de colas son:

- **insertar(Trabajo)**: Inserta un trabajo en el sistema de colas, si el trabajo es de prioridad ALTA irá a la cola denominada CPA, sino a la cola denominada CPN
- **extraer ()**: Extrae el trabajo a ejecutar en función de la disciplina de cola empleada (en este caso FIFO) y considerando que **se deben sacar primero los trabajos con prioridad alta**. Un trabajo cuando es sacado desaparece del sistema
- **estado()**: Devuelve el número de mensajes dentro del sistema de colas

Se pide

- Diseñe mediante una clase el sistema anterior y realice una representación en UML
- Implemente en C++ la clase anterior junto con la clase Mensaje
- Realice la siguiente prueba mediante un programa principal
 - Cree una instancia del sistema de colas
 - Cree tres mensajes: m1("OP1",prioridad =alta), m2 ("OP2",prioridad =alta), m3("OP3",prioridad =normal)
 - Envíe al sistema multi-cola los tres trabajos en el orden anterior
 - Imprima por pantalla el estado actual del sistema de colas
 - Ejecute la operación extraer **dos veces** del sistema de colas
 - Imprima por pantalla el estado actual del sistema de colas

Práctica 3. Java

Dentro de la red la Internet, encontramos múltiples dispositivos encargados de gestionar el transporte de la información entre los nodos de la red. Los enrutadores o encaminadores (Routers) se encargan de gestionar las múltiples rutas que puede seguir esta información. Un Router es un elemento de red **que decide que enlace va a seguir el datagrama**. Queremos modelar **de manera muy simple** el comportamiento de este componente. Para nosotros un **Router** tendrá las siguientes **características**:

- **Dos enlaces de salida.**
 - Enlace 1="193.146.145"
 - Enlace 2="193.132.128"
- **Un buffer** que permite almacenar temporalmente los datagramas entrantes. Con una **capacidad limitada** a N datagramas IP.

Las **operaciones** para esta clase serán las siguientes:

| Función | Descripción |
|------------------------|---|
| almacenar(DatagramaIP) | Sólo se aceptarán datagramasIP si DataGramalP direcc_Red_Destino=(Enlace 1 o Enlace2) o existe capacidad de almacenamiento suficiente NOTA: devuelve false si no se ha podido almacenar |
| encaminar | Por cada datagramaIP en el buffer Si DataGramalP direcc_Red_Destino =Enlace 1 encaminar por el enlace uno Si DataGramalP direcc_Red_Destino =Enlace2 encaminar por el enlace dos Sino error interno El datagramaIP cuando es encaminado cambia su valor de TTL a TTL-1 Nota: IMPRIMA POR PANTALLA el resultado del enrutamiento para poderlo verlo después en las pruebas |
| estado de Buffer | Numero de datagramas en el buffer |
| constructor | Parámetros → Enlaces de salida y capacidad máxima |

IMPORTANTE

Represente las **direcciones de red de los enlaces como cadenas**.

La **capacidad** se puede indicar con una constante de la siguiente forma

```
private final int CAPACIDAD=3;
```

Para representar el buffer use **el tipo abstracto Vector de Java SDK**

Se pide:

1. Diagrama UML de la clase Router
2. Realice la siguiente prueba sobre la clase Router
 - o Crear un Router con la siguiente configuración(Enlace 1="193.146.145",Enlace 2="193.132.129",CAPACIDAD=10)
 - a. Crear tres datagramas con los siguientes datos

| | IPorigen | IPdestino | TTL | Longitud total | DF |
|------------|----------------|----------------|-----|----------------|----|
| Datagrama1 | 193.123.134.34 | 193.130.129.34 | 3 | 35000 | 1 |
| Datagrama2 | 193.123.134.34 | 193.146.145.56 | 2 | 34000 | 0 |
| Datagrama3 | 193.123.134.34 | 193.130.129.58 | 3 | 32000 | 1 |
| | | | | | |

- b. Almacene cada datagrama en el Router
- c. Obtenga el valor de estado del Router
- d. Invocar a la operación encaminar() del Router
- e. Obtenga el valor de estado
- f. Cree ahora una prueba donde se sature la capacidad del router