

**Centro Universitario de Ciencias
Exactas e Ingenierías**

Universidad de Guadalajara

REPORTE 04

Algoritmo de planificación FCFS

Continuación

Alumnos:

Carbajal Armenta Yessenia Paola
Sánchez Lozano Jonathan

Códigos:

220286482
215768126

Profesora:

Becerra Velázquez Violeta del Rocío

Materia:

Seminario de Soluciones de
Problemas de Sistemas Operativos

Departamento:

Ciencias Computacionales

Carrera:

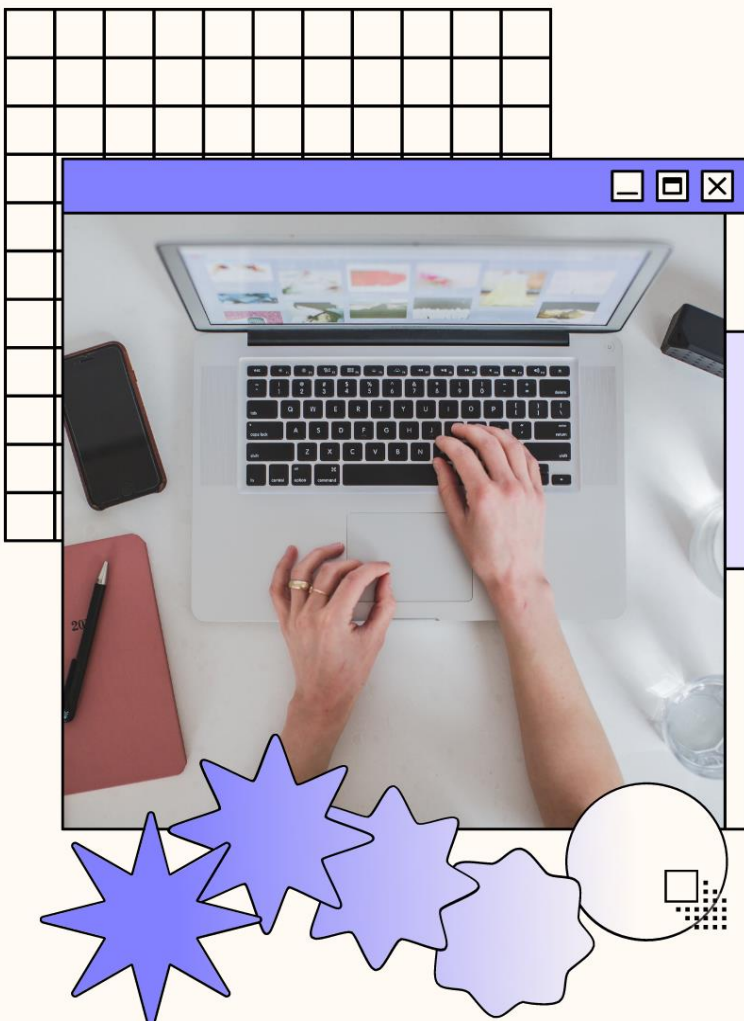
Ingeniería en Computación

NRC:

103844

Sección:

D01



Actividad No. 8

Algoritmo de planificación FCFS (First Come First Server)

Continuación

Introducir desde teclado N procesos, estos serán los que conformen el número de procesos a terminar llevando a cabo la implementación de un algoritmo de planificación FCFS, el cual consiste en ejecutar automáticamente aplicaciones y procesos de cola en orden de llegada.

Objetivo

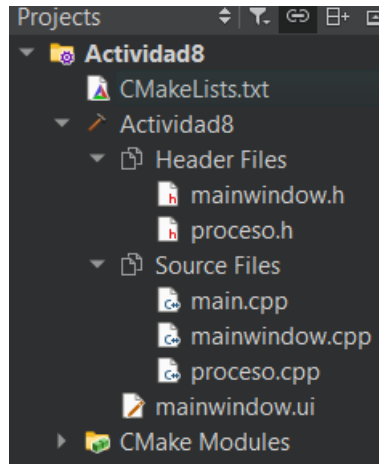
Para este programa respecto al último que realizamos, es que se continúa con todo lo que tiene el anterior, con las novedades que agregamos que se puede agregar un nuevo proceso, se muestra una tabla de procesos y si se pausa el programa o se manda a la tabla de procesos no se puede hacer nada hasta que se le dé a la tecla de continuar.

Desarrollo

Para el desarrollo de la práctica decidimos usar el lenguaje de programación C++ en complemento con el framework QT. Seleccionamos este lenguaje ya que, además de ser con el que más cómodo nos sentimos, nos da un campo más amplio a la hora de manejar con instrucciones de nivel un poco más bajo, en comparación de Python u otros lenguajes de nivel superior que tienden a obviar algunas partes.

Se desarrolló el programa con programación orientada a objetos y con la funcionalidad de una interfaz gráfica para una mayor comodidad, se crearon instancias para cada para cada proceso y así poder almacenarlos por lotes para facilitar su proceso y así poder simularlo de una manera más real.

Se terminaron creando los siguientes archivos:



Al igual que en el programa pasado, seguimos empleando C++ y el framework de Qt ya que, pese a las complicaciones presentadas en las actividades previas, consideramos que hemos aprendido a utilizarlos de mejor manera, además, llegados a este punto, pese a que cada programa puede ser abordado desde el principio con algún otro lenguaje y/o framework sin problemas.

Para esta práctica en específico, se le añadieron las funciones necesarias para hacer posible el uso de las teclas requeridas para esta práctica.

Para validar las nuevas pulsaciones de las teclas, se añadieron las opciones de 'N' como nuevo proceso y 'B' como tabla de procesos al sistema de reconocimiento de teclado:

```
else if ( event->key() == Qt::Key_N and state == RUNNING ){
    qDebug() << "Nuevo";
    createNewProcess();
}
else if ( event->key() == Qt::Key_B and state == RUNNING ) {
    qDebug() << "Tabla de procesos";
    state = PAUSED;
    ui->stackedWidget->setCurrentIndex(SHOW_TIMES_PROCESSES);
    showProcessesTimes();
    this->setFocus();
    pause.exec();
    this->setFocus();
}
```

En la tabla de procesos se muestra hasta el final de tabla en qué estado se encuentra el proceso:

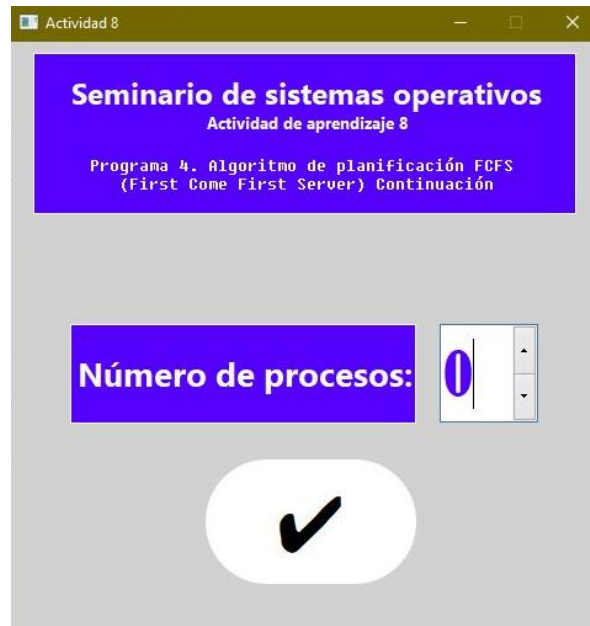
```
switch(p.estado) {
    case States::Finished:
    case States::Ready:
    case States::Blocked:
    case States::Executing:
        arrived = QString::number(p.getTiempoLlegada());
        service = QString::number(p.getTiempoServicio());
        operation = QString(p.getOperacion().c_str());
        waiting = QString::number(p.getTiempoEspera());
        if ( p.getTiempoBloqueado() != 0 ) {
            locked = QString::number(7-p.getTiempoBloqueado());
        } else {
            locked = "0";
        }
        if ( tr != 0 ) {
            remaining = QString::number(p.getTiempoEstimado()-p.getTiempoTranscurrido());
        } else {
            remaining = "0";
        }
    }

    switch(p.estado) {
        case States::Finished:
            finished = QString::number(p.getTiempoFinalizacion());
            return = QString::number(p.getTiempoRetorno());
            request = QString::number(p.getTiempoRespuesta());
            if ( p.getFinalizacion() ) {
                result = QString::number(p.getResultadoOperacion());
            } else {
                result = finished = return = arrived = service = request = waiting = locked = remaining = operation ="ERROR";
            }
            break;
        default:
            result = finished = return = request = "NULL";
    }
    break;
}

default:
    result = finished = return = arrived = service = request = waiting = locked = remaining = operation ="NULL";
    break;
}

switch(p.estado) {
    case States::Ready: stateProcess = "LISTO"; break;
    case States::New: stateProcess = "NUEVO"; break;
    case States::Executing: stateProcess = "EJECUCIÓN"; break;
    case States::Blocked: stateProcess = "BLOQUEADO"; break;
    case States::Finished: stateProcess = "FINALIZADO"; break;
    default: ;
}
```

Programa en ejecución:



Actividad 8

Datos de los procesos

ID	Estado	T. Bloqueado	Operación	Resultado	T. Llegada	. Finalizaci	T. Retorno	T. Espera	T. Servicio	T. Restante	T. Respuest
1	FINALIZADO	0	49°66	-2147483648	0	27	27	13	14	0	0
2	FINALIZADO	ERROR	ERROR	ERROR	ERROR	ERROR	ERROR	ERROR	ERROR	ERROR	ERROR
3	FINALIZADO	0	10/-14	0	3	34	31	19	12	0	8
4	BLOQUEADO	4	85+-33	NULL	11	NULL	NULL	34	6	10	NULL
5	BLOQUEADO	6	59°21	NULL	27	NULL	NULL	38	2	12	NULL
6	EJECUCIÓN	0	-73+78	NULL	34	NULL	NULL	39	1	6	NULL
7	NUEVO	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Actividad 8

Datos de los procesos

ID	Estado	T. Bloqueado	Operación	Resultado	T. Llegada	. Finalizaci	T. Retorno	T. Espera	T. Servicio	T. Restante	T. Respuesta
1	FINALIZADO	0	49°66	-2147483648	0	27	27	13	14	0	0
2	FINALIZADO	ERROR	ERROR	ERROR	ERROR	ERROR	ERROR	ERROR	ERROR	ERROR	ERROR
3	FINALIZADO	0	10/-14	0	3	34	31	19	12	0	8
4	FINALIZADO	0	85+-33	52	11	56	45	29	16	0	5
5	FINALIZADO	0	59°21	-2147483648	27	68	41	27	14	0	10
6	FINALIZADO	0	-73+78	5	34	46	12	5	7	0	5
7	FINALIZADO	0	60°68	-2147483648	46	82	36	22	14	0	22

Conclusiones

Carbajal Armenta Yessenia Paola:


Esta práctica ha sido de las más fáciles hasta el momento, ya que de lo que teníamos de la anterior práctica solo fue agregar unas pequeñas cosas que fue

agregar un nuevo proceso, mostrar la tabla de los procesos al momento, mostrar el estado de los procesos, entre otras cosas.


Sánchez Lozano Jonathan:

En los personales, esta práctica ha sido la más sencilla de realizar de las 4 que llevamos por el simple hecho que solamente fue agregar unas cuantas cosas al programa anterior, solamente tuvimos unos imprevistos con el acomodo de los tiempos, pero fuera de eso todo salió de manera correcta. Cabe destacar que la parte de agregar un proceso nuevo y mostrar la tabla de los procesos al momento, fue más sencillo de lo que pude imaginar en un principio.

Enlace del código

 <https://drive.google.com/drive/folders/1i1Wmz3FSas6WnEJa-SQfGjm2yF13Q9PS?usp=sharing>

Enlace del vídeo

 <https://drive.google.com/file/d/1iPs4Oyf9RBCgKuN0O00DEWerabWp2p6I/view?usp=sharing>