

**Centro Universitario de Ciencias  
Exactas e Ingenierías**

**Universidad de Guadalajara**

## **REPORTE 05**

### **Algoritmo de planificación**

#### **Round Robin**

#### **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:**

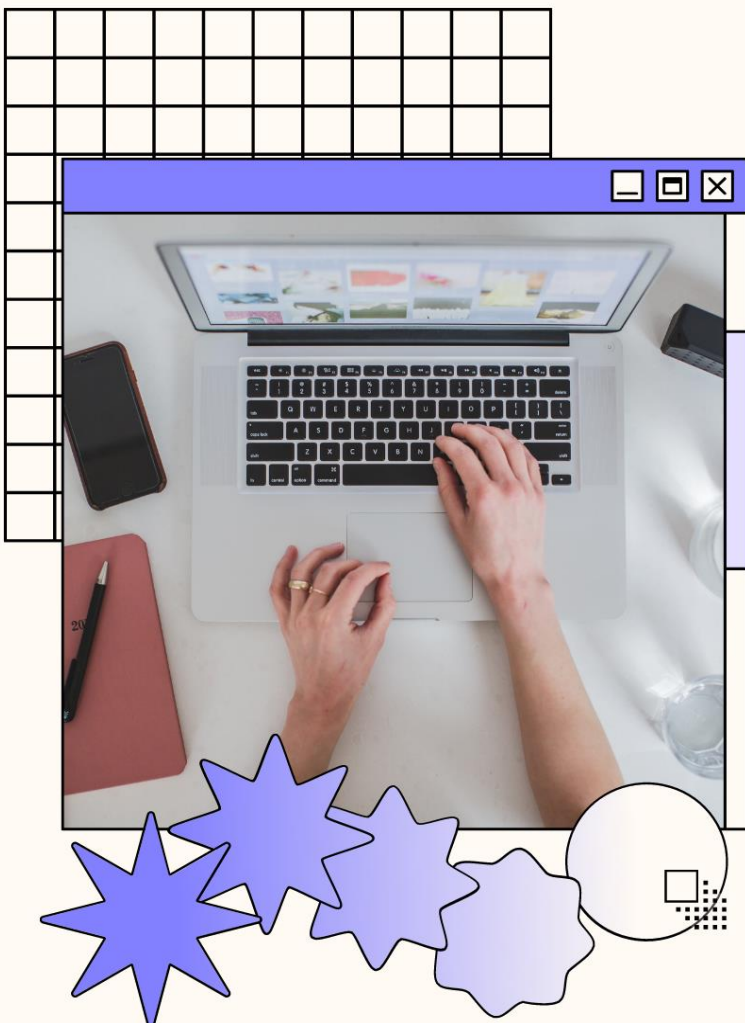
Ingeniería en Computación

#### **NRC:**

103844

#### **Sección:**

D01



---

## *Actividad No. 10*

---

### **Algoritmo de planificación Round Robin**

Introducir desde teclado N procesos, estos serán los que conformen el número de procesos a terminar. Se cumplirán con todas las especificaciones de la práctica anterior, con excepción del algoritmo de planificación, se manejará la cola de listos utilizando el algoritmo de planificación Round-Robin.

---

### *Objetivo*

---

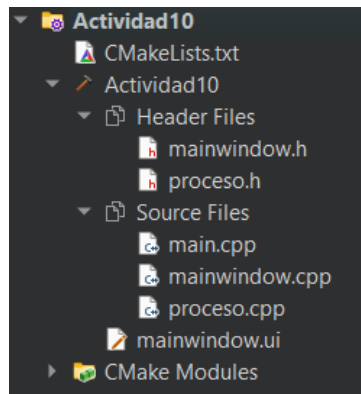
Para este programa respecto al último que realizamos, se usó la implementación del algoritmo de planificación de Round Robin y solo se le añadió el poder poner un número de Quantum para cada proceso y todo lo demás quedó prácticamente igual, solo con la modificación de la manera de trabajo de los procesos en ejecución simulando un carrusel.

---

### *Desarrollo*

---

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. 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 tecleo:

### ***Mainwindow.h***

En este apartado se agregó en la última línea visible el valor que se le asignará al quantum en la ejecución del programa.

```
63  private slots:
64
65      void on_processNumberSP_valueChanged(int arg1);
66      void on_processAccountPB_clicked();
67      void on_startRunPB_clicked();
68      void enable_initialButton();
69      void on_quantumSP_valueChanged( int arg1 );
```

### ***Proceso.h***

En esta sección se define de manera privada el quantum y de manera pública el quantumaux para evitar problemas de implementación y además se agregó el setter y getter del quantum.

```

46 private:
47     int id;
48     string operacion;
49     int resultadoOperacion;
50     bool finalizacion;
51     bool ejecutado;
52     int quantum;
53     int times[Times::Count] = {0};
54
55 public:
56     int quantumaux;

```

```

102     int getQuantum() const;
103     void setQuantum(int value);

```

### Proceso.cpp

En este archivo se agregaron las siguientes funciones de get y set Quantum para definir los métodos de set y get.

```

178 int Proceso::getQuantum() const {
179     return quantum;
180 }
181
182 void Proceso::setQuantum(int value) {
183     quantum = value;
184 }

```

En esta sección se agregó la impresión en las tablas del quantum, el cual se va incrementando de manera continua hasta llegar al valor asignado.

```

223 if ( readyProcesses[0].getId() != NULL_PROCESS ) {
224
225     readyProcesses[0].quantumaux=0;
226 while (tT < readyProcesses[0].getTiempoEstimado() and tT != ACTION_CODE ){
227     tR--;
228     tT++;
229     readyProcesses[0].setTiempoTranscurrido(tT);
230     readyProcesses[0].quantumaux++;
231
232     ui->processRuningTB->setItem(0,0,new QTableWidgetItem(QString::number(readyProcesses[0].getId())));
233     ui->processRuningTB->setItem(1,0,new QTableWidgetItem(QString(readyProcesses[0].getOperacion().c_str())));
234     ui->processRuningTB->setItem(2,0,new QTableWidgetItem(QString::number(readyProcesses[0].getTiempoEstimado())));
235     ui->processRuningTB->setItem(3,0,new QTableWidgetItem(QString::number(readyProcesses[0].getTiempoTranscurrido())));
236     ui->processRuningTB->setItem(4,0,new QTableWidgetItem(QString::number(tR)));
237     ui->processRuningTB->setItem(5,0,new QTableWidgetItem(QString::number(readyProcesses[0].quantumaux)));
238
239 if (readyProcesses[0].quantumaux == readyProcesses[0].getQuantum()){
240     if (readyProcesses.size() > 1){
241         Proceso p = readyProcesses[0];
242         p.quantumaux=0;
243         readyProcesses[0].quantumaux=0;
244         p.estado = States::Ready;
245         readyProcesses.push_back(p);
246         readyProcesses.erase(readyProcesses.begin());
247         tT = readyProcesses[0].getTiempoTranscurrido();
248         tR = readyProcesses[0].getTiempoEstimado() - tT;
249         showReadyProcesses();
250     }else{
251         readyProcesses[0].quantumaux=0;
252     }
253 }

```

### Programa en ejecución:

Actividad 10

Seminario de sistemas operativos

Actividad de aprendizaje 10

Programa 5. Algoritmo de planificación

Round Robin

Número de procesos:

0

Valor del Quantum:

0

✓

Actividad 10

Cola de listos

ID	M. Estimac	T. Transcurrido
----	------------	-----------------

Procesos nuevos:

0

Tiempo Quantum:

3

Procesos terminados

ID	Operación	Resultado
----	-----------	-----------

Tiempo transcurrido:

0

Proceso en ejecución

ID	Operación
	T. M. Estimado
	T. Transcurrido
	T. Restante
	Quantum

Cola de bloqueados

ID	T. en bloquea
----	---------------

Simular

### 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	ERROR	16%14	ERROR	ERROR	ERROR	ERROR	ERROR	ERROR	ERROR	ERROR
2	FINALIZADO	0	-86%-87	-86	0	39	39	27	12	0	12
3	FINALIZADO	ERROR	-95+54	ERROR	ERROR	ERROR	ERROR	ERROR	ERROR	ERROR	ERROR
4	FINALIZADO	0	15-99	-84	5	36	31	23	8	0	0
5	FINALIZADO	0	1+26	27	12	33	21	15	6	0	0
6	FINALIZADO	0	32~46	-2147483648	33	70	37	23	14	0	3
7	FINALIZADO	0	54%-4	2	36	73	37	25	12	0	3
8	FINALIZADO	ERROR	25*-46	ERROR	ERROR	ERROR	ERROR	ERROR	ERROR	ERROR	ERROR
9	FINALIZADO	0	37/-31	-1	65	85	20	9	11	0	5
10	FINALIZADO	ERROR	-41-77	ERROR	ERROR	ERROR	ERROR	ERROR	ERROR	ERROR	ERROR

### Conclusiones


#### Carbajal Armenta Yessenia Paola:

Fue una actividad demasiado sencilla por el simple hecho de que fueron muy pocas líneas de código las que se agregaron para poder desarrollar este entregable, fue bastante interesante ver también la implementación de este algoritmo de planificación.

#### Sánchez Lozano Jonathan:

Esta actividad hasta el momento ha sido de las más sencillas que hemos hecho ya que solo fue agregar unas pequeñas cosas y no supuso un gran reto porque no tardamos mucho en hacerlo, además, me pareció muy interesante el uso y la implementación del quantum para este programa.

### Enlace del código

 [https://drive.google.com/drive/folders/1iYdXjhuv8OhgCFxnADYzF\\_BMPWLCAL94?usp=sharing](https://drive.google.com/drive/folders/1iYdXjhuv8OhgCFxnADYzF_BMPWLCAL94?usp=sharing)

---

*Enlace del vídeo*

---

 [https://drive.google.com/file/d/17ofpg7CP4yOVmvT6yRGSTOfHFN\\_xWnIH/view?usp=sharing](https://drive.google.com/file/d/17ofpg7CP4yOVmvT6yRGSTOfHFN_xWnIH/view?usp=sharing)