

**UNIVERSIDAD DE GUADALAJARA**

Centro Universitario De Ciencias Exactas E Ingenierías

División de Tecnologías para la Integración Ciber-Humana

Seminario de Solución de Problemas de Sistemas Operativos I7030

Reporte de Actividad Práctica

**Actividad 4 . Administrador de procesos**

Alumnos:

Hernández Ledezma Brandon 215515031

Martínez Castillo Lisseth Abigail 218292645

Ramírez Plascencia Miguel Alejandro 215628413

Profesor: Quintanilla Moreno Francisco Javier

11 de septiembre de 2022

## Multiprogramación

### Objetivo de la práctica

El alumno identificará y diferenciará los diferentes estados y comportamientos de los procesos en un sistema operativo demostrando sus conocimientos por medio de un programa que actúe de forma similar.

### Antecedentes

#### ¿Qué es un proceso?

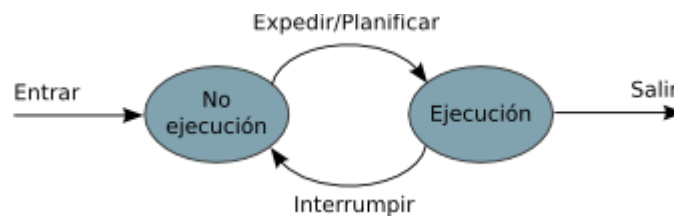
Un proceso es la “imagen en memoria de un programa”. Un proceso es una entidad activa que emplea un programa (lista de instrucciones) para definir cómo actuará el sistema. En otras palabras es un concepto manejado por el sistema operativo que consiste en el conjunto formado por:

- Las instrucciones de un programa destinadas a ser ejecutadas por el microprocesador.
- Su estado de ejecución en un momento dado, esto es, los valores de los registros de la CPU para dicho programa.
- Su memoria de trabajo, es decir, la memoria que ha reservado y sus contenidos.
- Otra información que permite al sistema operativo su planificación.

La responsabilidad principal del sistema operativo es controlar la ejecución de los procesos. Esto incluye determinar el patrón de entrelazado para la ejecución y asignar recursos a los procesos. El primer paso en el diseño de un sistema operativo para el control de procesos es describir el comportamiento que se desea que tengan los procesos.

### Modelo de procesos de dos estados

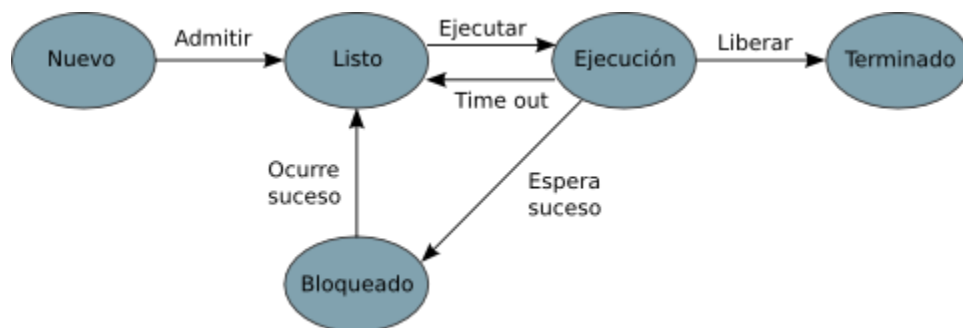
Se puede construir el modelo más simple posible, observando que en un instante dado, un proceso está siendo ejecutado por el procesador o no. En este modelo, un proceso puede estar en dos estados: **Ejecutando** o **No Ejecutando**. Cuando el sistema operativo crea un nuevo proceso, crea el bloque de control de proceso para el nuevo proceso e inserta dicho proceso en el sistema en estado No Ejecutando. El proceso existe, y es conocido por el sistema operativo, y está esperando su oportunidad de ejecutar.



## Modelo de proceso de cinco estados

Una forma óptima para manejar los procesos es dividir el estado en No Ejecutando en dos estados: Listo y Bloqueado. Para gestionarlo correctamente, se han añadido estados adicionales que serán bastante útiles. Estos cinco estados son:

- **Ejecutando:** El proceso está actualmente en ejecución.
- **Listo:** Un proceso que se prepara para ejecutar cuando tenga oportunidad.
- **Bloqueado:** Un proceso que no puede ejecutar hasta que se cumpla un evento determinado o se complete una operación E/S.
- **Nuevo:** Un proceso que se acaba de crear y que aún no ha sido admitido en el grupo de procesos ejecutables por el sistema operativo. Suele ser un proceso que aún no se ha cargado en memoria principal aunque su BCP sí ha sido creado.
- **Terminado:** Un proceso que ha sido liberado del grupo de procesos ejecutables por el sistema operativo, debido a que ha sido detenido o que ha sido abortado por alguna razón.



Estos son los tipos de eventos que llevan a cada transición de estado para cada proceso:

- **Null -> Nuevo:** Se crea un proceso nuevo para ejecutar un programa.
- **Nuevo -> Listo:** El sistema operativo mueve a un proceso del estado Nuevo al estado Listo cuando éste se encuentre preparado para ejecutar un nuevo proceso. La mayoría de sistemas fijan un límite basado en el número de procesos existentes o en la cantidad de memoria virtual que se podrá utilizar por parte de los procesos existentes. Este límite garantiza que no haya demasiados procesos activos y que se degrade el rendimiento del sistema.
- **Listo -> Ejecutando:** Cuando llega el momento de seleccionar un nuevo proceso para ejecutar, el sistema operativo selecciona uno de los procesos que se encuentren en estado Listo. Esta tarea la realiza el planificador.
- **Ejecutando -> Saliente:** El proceso que actualmente está en ejecución se finaliza por parte del sistema operativo, tanto si ha finalizado su ejecución o si ha sido abortado por algún motivo.

- **Ejecutando -> Listo:** Esta transición se suele producir cuando el proceso que está en ejecución ha alcanzado el máximo tiempo posible de ejecución de forma ininterrumpida. Prácticamente todos los sistemas multiprogramados imponen este tipo de restricción de tiempo.
- **Ejecutando -> Bloqueado:** Un proceso se pone en estado Bloqueado si solicita algo por lo que debe esperar. Pueden ser muchos los motivos de esta transición: solicitar una operación de E/S, solicitar una sección de memoria compartida que no está inmediatamente disponible, comunicarse con otro proceso y bloquearse hasta que el otro proceso le proporcione los datos solicitados, etc.
- **Bloqueado -> Listo:** Un proceso en estado Bloqueado se mueve a Listo cuando sucede el evento por el cual esperaba.
- **Listo -> Saliente:** Para simplificar, esta transición no se muestra en la imagen del modelo de cinco estados. Un padre puede terminar la ejecución de un proceso hijo en cualquier momento. De igual manera, si el padre termina, todos los procesos hijos asociados al padre pueden finalizarse.
- **Bloqueado -> Saliente:** Igual que el caso anterior.

## Desarrollo

Para el desarrollo de esta práctica se implementó una simulación donde se muestra cómo al iniciar la ejecución del programa los procesos están en blanco y hay distintos botones que inician, terminan, pausan y des pausan simulando los estados por los que pasan los procesos.

```
taskManager.py X main.py
taskManager.py > ...
26
27 def stopThread(widget, index):
28     widget.thread[index].stop()
29
30 def currentValues(widget, index):
31     bars = [widget.bar_0,
32             widget.bar_1,
33             widget.bar_2,
34             widget.bar_3]
35     value=bars[index].value()
36     return value
37
38 def startThread(widget, index):
39     random=randint(10, 500)
40     value=currentValues(widget, index)
41     widget.thread[index]=threadClass(parent=None, index=index, value=value)#asignamos a una posicion del arreglo un objeto de la clase threadClass que seran
42     #nuestros hilos y el objeto adquiere en su atributo index su posicion dentro del arreglo a manera de identificacion
43     QTest.QTest.qWait(random)#esperamos un determinado tiempo
44     widget.thread[index].start()#señalizamos que nuestro hilo iniciara su proceso
45     widget.thread[index].any_signal.connect(widget.taskManager)#
46
47
```

Imagen 2.1 Código del desarrollo del algoritmo para la simulación del administrador de procesos.

```
taskManager.py  main.py x
main.py > ...
22 def pauseOne():
23     index=int(playButtons.index(widget.sender()))
24
25     startButtons[index].setEnabled(True)
26
27
28     if(playButtons[index].text()=="|||"):
29         playButtons[index].setText("▶")
30         setFinished(status, index)
31         stopThread(widget, index)
32
33     elif(playButtons[index].text()=="▶"):
34         playButtons[index].setText("|||")
35         setRunning(status, index)
36         startThread(widget, index)
37
38
39
40 def stopOne():
41     index=int(endButtons.index(widget.sender()))
42     endButtons[index].setEnabled(False)
43     startButtons[index].setEnabled(True)
44     playButtons[index].setText("▶")
45     playButtons[index].setEnabled(False)
46     setFinished(status, index)
47     bars[index].setValue(0)
48     stopThread(widget, index)
49
50
51 def startOne():
52     index=int(startButtons.index(widget.sender()))
53     endButtons[index].setEnabled(True)
54     startButtons[index].setEnabled(False)
55     playButtons[index].setEnabled(True)
56     playButtons[index].setText("|||")
57     print(index)
58     setRunning(status, index)
```

Imagen 2.2 Código del desarrollo del algoritmo para la simulación del administrador de procesos.

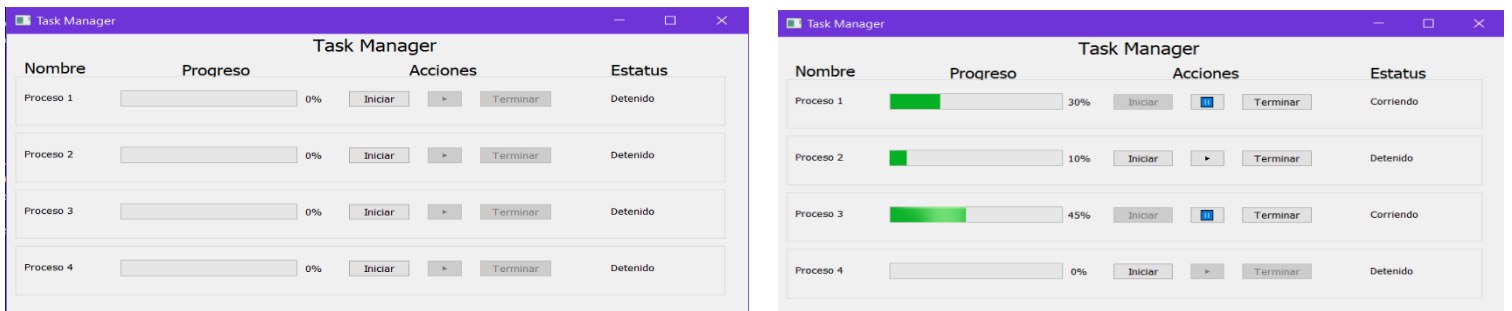


Imagen 3 Imagen donde es posible visualizar la simulación del task manager, cada barra tiene botones que controlan los estados del proceso.

## **Conclusión:**

### **Brandon Hernández Ledezma:**

Como se ha visto a lo largo de las clases los sistemas operativos están encargados de controlar todo por medio de procesos, para esto el sistema operativo necesita de algoritmos para poder intercambiar recursos entre distintos procesos. Con este fin a los procesos se les doto de estados los cuales le permiten a un administrador de procesos poder hacer distintas acciones con ellos, en este caso son 5 los estados que puede tomar un proceso: Nuevo, Listo, En ejecución, Bloqueado, Terminado. Los cuales se van intercambiando, dependiendo el caso para que los distintos procesos que se encuentran también procesándose puedan trabajar de manera eficiente en un ambiente en el cual se tienen que ejecutar varias tareas a la vez. Con esta practica me quedo mucho mas claro cómo es que funcionan los estados de los procesos y las transiciones que puede haber entre ellos.

### **Liseth Abigail Martínez Castillo:**

Llegado a este punto, hemos aprendido como el sistema operativo se encuentra estructurado en una primera etapa, el desarrollo de procesos y la comprensión de los hilos de ejecución a través de los cuales, el sistema delega funciones y opera en forma multi funcional. También estudiamos la forma en la que el sistema gestiona las interrupciones, este se encarga de controlar los accesos al procesador, verificar el estatus de un proceso y determinar se ejecución de acuerdo con el nivel de importancia. La manera en la que opera nuestra computadora es muy compleja, una vez más comprender su funcionamiento, nos permite prever soluciones a ciertos problemas de ejecución

### **Miguel Alejandro Ramírez Plascencia:**

Los procesos son una parte crucial del funcionamiento de cualquier programa en nuestros equipos. Normalmente, estos procesos están coordinados por el sistema operativo, encargado de gestionar los recursos empleados por el sistema; unos recursos muy limitados y de uso constante, por lo que los procesos suelen moverse entre varios estados para acaparar los recursos el mínimo tiempo posible. Los procesos se encuentran en diferentes estados según las necesidades del equipo, sus recursos, y la coordinación y manipulación por parte del S.O. De estos distintos estados, los más básicos serían: Nuevo, Activo, Preparado, Bloqueado y Terminado. Encontrándose obligatoriamente en alguno de ellos desde el comienzo del proceso a la finalización del mismo.

## **Fuentes:**

- Gibran García. (2020, 14 octubre). *Entendiendo administración de procesos en Sistemas Operativos*. Naps Tecnología y educación. Recuperado 10 de septiembre de 2022, de <https://naps.com.mx/blog/administracion-de-procesos-en-sistemas/>
- Sistemas operativos – Estados de los procesos. (2015, 23 julio). THEBITDEVELOPER. Recuperado 10 de septiembre de 2022, de <https://thebitdeveloper.com/sistemas-operativos-estados-de-los-procesos/>
- 2.1 Concepto de proceso. (s.f.). Recuperado 10 de septiembre de 2022, de <https://sites.google.com/site/materiasisoperativo/unidad-2-administrador-del-proceso-y-del-procesador/2-1-concepto-de-proceso>