2020

Manual de Estándares

INTEGRANTES

1) Gerson Orlando Castillo Arce ---------->20171930094
2) Xenia Larissa Alfaro Núñez ----------> 20171030033
3) Carmen María Cárcamo ---------->20171000552
4) Katherine Arely Espino Santos ---------->20171001091

Índice

Introducción

El documento a desarrollar a continuación explica el desarrollo del modelo de cinco estados y sus transiciones, se muestra la estructura, el diseño y la descripción de cada proceso. El proyecto fue desarrollado en el lenguaje de JavaScript usando el entorno de desarrollo Visual Studio Code, el diseño de la estructura fue desarrollado en HTML y Css, usando el Framework Bootstrap para poder representar el bloque de control de procesos que tiene almacenado la información de cada uno de los procesos a ejecutar.

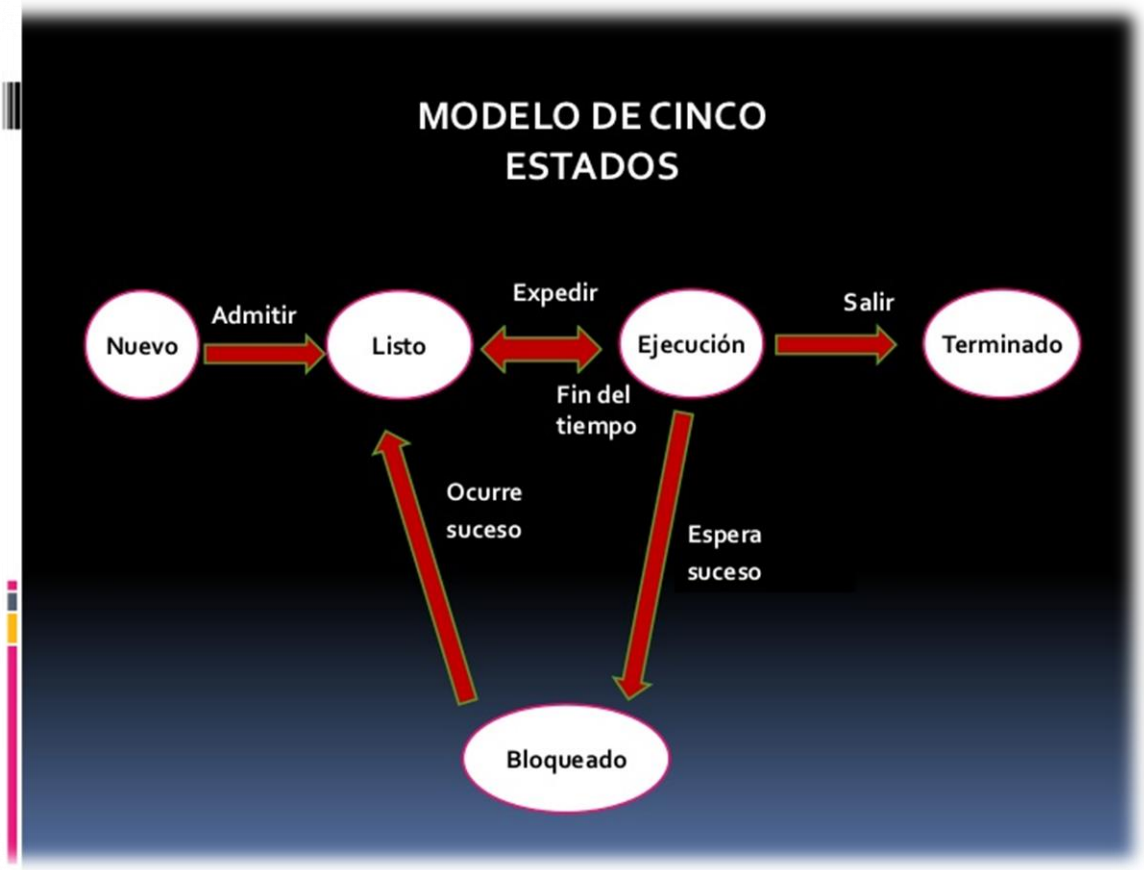
Objetivos

1- El principal objetivo del simulador de gestor de procesos es conocer el funcionamiento y la estructura del modelo de cinco estados.

2- Simular la gestión de procesos en un procesador.

3- Representar de manera gráfica el funcionamiento de un gestor de un proceso.

Modelo de Cinco Estados



Descripción Del Simulador de Gestor de Proceso

Estado	Descripción
Nuevo	En este estado el proceso recién fue creado , se guarda en una respectiva cola sin importar la prioridad que tenga dicho proceso , cada uno de ellos espera para que sea despachado al otro estado de listo donde en ese estado es depende de la prioridad que contenga cada uno de los procesos .
Listo	En este estados guarda los procesos según sea su prioridad, dispone de 3 colas donde cada vez que vengan los procesos de nuevo se almacenen de acuerdo a la prioridad y así poder esperar hasta que pase al estado de ejecución.
Ejecución	En este estado el proceso está actualmente en ejecución, este es un estado que depende de la prioridad, y ejecuta la prioridad más alta que tenga el proceso en este caso la prioridad más alta es 1 y la más baja 3.
Bloqueado	En este estado el proceso no se puede ejecutar hasta que no se produzca ciertos sucesos en este caso los eventos que será la causa de bloqueo de un proceso es: Evento entrada y salida (3) e disco duro (5).
Terminar	En este estado el proceso fue expulsado del grupo del proceso ejecutable ya sea porque termino o por algún fallo.

Diccionario de Datos

Nombre Variable	Tipo de Dato	Descripción
cNuevo	VAR	Cola de estado nuevo
cListoP1	VAR	Cola de estado listo para procesos con prioridad 1
cListoP2	VAR	Cola de estado listo para procesos con prioridad 2
cListoP3	VAR	Cola de estado listo para procesos con prioridad 3
cTermiando	VAR	Cola de estado terminado
cBloqueado3	VAR	Cola de estado bloqueado para evento 3
cBloqueado5	VAR	Cola de estado bloqueado para evento 5
Ejecutando	VAR	Variable que indica si el gestor está ejecutando un proceso

Nombre Variable	Tipo de Dato	Descripción
Despachando	VAR	Variable que indica si el gestor está llevando a cabo un cambio de estado.
Temporizador	VAR	Variable que lleva los ciclos del temporizador
idUltimoEjecutado	VAR	Variable utilizada en evitar la monopolización de procesos
ContadorEjecuciones	VAR	Variable utilizada en evitar la monopolización de procesos.
TextoADocumento	VAR	Variable de tipo string para concatenar la versión string de los procesos que se ejecutan
salto	VAR	Variable que almacena los saltos ingresados por el usuario.
NProcesos	VAR	Numero de procesos que se gestionara , obtenidola de forma aleatoria

Clases del programa

Clase	Descripción
Queue	La instancia de esta clase nos permite la construcción de los objetos de tipo cola
Procesos	Clase 'Proceso' para poder instanciar y hacer uso de un proceso y de sus atributos.
NodeQ	Clase 'Queue' para poder instanciar y hacer uso de una cola

Atributos de las clases

Queue:

Atributo	Descripción
First	Primer elemento de la cola
Array	Arreglo que contiene los elementos de la cola

Proceso

Atributos	Descripción
Id	Id de proceso
Estado	Estados del proceso : nuevo ,listo, ejecutando, Bloqueado, Terminado
Prioridad	Prioridad que contiene el proceso en este caso sería: (1), (2), (3). Siendo la prioridad más alta “1 “.
CantInst	Cantidad de instrucciones del proceso.
InstBloq	Numero de instrucciones , donde se realizó el bloqueo
Evento	Evento que genera el bloqueo pueden ser : 3 =E/S 5=Disco duro
PC	Contador del programa
ContadorEvento	Si el proceso está bloqueado el contador de evento lleva los ciclos ejecutados del evento bloqueado

aString	Concierte a una cadena todos los datos del proceso

Funciones del programa

- **function iniciar() :** Se ejecuta solo al inicio para crear los procesos y actualizar las colas que se imprimen.
- **function verificarColas():** Concierte a una cadena todos los datos del proceso Esta función se manda a llamar cada vez que se apretar el botón "avanzar" para deshabilitarlo si todos los procesos han terminado de ejecuta sus instrucciones.
- **function crearProcesos():** Obtiene el número de procesos que va a contener el gestor de procesos de manera aleatoria. Crea una instancia por cada proceso y se le asignan los valores a los atributos (algunos de manera aleatoria) para después agregar ese proceso a la cola de nuevo.
- **function randomN ():** Genera los número aleatoriamente.
- **function completar(numero, longitud):** completa con ceros la cantidad de carácteres establecido para los atributos del proceso que así lo requieran.
- **function actualizarVentana():**Se encarga de actualizar las colas que se imprimen en pantalla obteniendo los "datos del proceso" de todos los procesos que hay en todas las colas.
- **function impresionCola():**Limpia cada cola que se imprime en pantalla y obtiene el string de todos los procesos que se encuentran en la cola que recibió como parámetro. Luego se manda a escribir dentro del cuadro que representa esa cola.
- **function imprimir**(): Imprime en consola "los datos de proceso" de los procesos que se han creado.
- **function avanzar():**Esta es la función principal. Se manda a llamar al darle click al botón de "avanzar" y dependiendo del estado de las colas y de la cuenta de ciclos ingresados por el usuario despacha o ejecuta procesos. También en esta función se concatenan "los datos de procesos" de cada proceso.
- **function vacias():**Esta función verifica que todas las colas (excepto la de terminado) esten vacías y la variable que contiene el proceso que se esta ejecutando(ejecutando) sea null(no se este ejcutando un proceso).
- **function procesarBloqueado():**Esta función primero verifica si hay procesos en la cola de bloqueados y de haberlos, revisar la cantidad de ciclos del evento que se han ejecutado para este. Si ya ha terminado la ejecución de la cantidad de ciclos correspondientes al evento entonces se saca el proceso de la cola de bloqueado para agregarlo a la cola de listo que corresponda. Realiza el cambio del atributo "estado" del proceso a 1(listo) de haberse realizado el cambio de cola.

- **function agregarListo():**Agrega el proceso a la cola de listo que corresponda (cListoP1 para prioridad 1, cListoP2 para prioridad 2, cListoP3 para prioridad 3). Realiza el cambio del atributo "estado" del proceso a 1(listo).
- **function procesarNuevos():**Cambie el proceso de la cola de nuevo a la de listo que corresponda (haciendo una llamada a la función "agregarListo()"). Realiza el cambio del atributo "estado" del proceso a 1(listo).
- **function procesarEjecutando():**Esta función verifica que si se esta ejecutando un proceso entonces se obtiene su pc para comparar este con la cantidad de instrucciones del proceso y la instrucción de bloqueo del proceso, para saber si se este debe terminar, o bloquearse respectivamente. Si la instrucción no corresponde a la de terminar proceso o a la de bloqueo, entonces el proceso se agrega a cola de listo para proseguir con su ejecución posteriormente.
- **function procesarListo():**se encarga de obtener el proceso que se va a ejecutar de la cola de listo. Luego evalúa si este proceso es igual al proceso * anteriormente ejecutado, de no ser así entonces este proceso se ejecutara. Si si es igual al anteriormente ejecutado el máximo de segmentos *seguidos que puede tener el proceso es 3. Si el proceso quiere ejecutarse por cuarta vez consecutiva, este baja de prioridad, se reingresa a la cola de listo correspondiente y se obtiene el próximo proceso con mayor prioridad para ejecutarse.
- **function cambiarCola():** Cambia de cola el proceso que recibe como parámetro, esto según la prioridad del proceso.
- **function ejecutarInstruccion():** ejecuta una instrucción del proceso que esta ejecutándose, incrementa el temporizador y el pc del proceso, concatena sus "datos del proceso" al string que contiene todos los procesos que se han ejecutado y por último verifica si se ha cumplido un segmento (temporizador==5), o se ha llegado al final de la ejecuión del proceso, o si se ha llegado a la instrucción de bloqueo para establecer despachando = true;
- **function contarEvento():**Ejecuta una ciclo del evento de bloqueo por cada uno de los elementos de las colas de bloqueado (cBloqueado3, cBloqueado5).
- **function Node(value):** Permite la construcción de los objetos tipo cola.
- **function Proceso ():** Clase 'Proceso' para poder instanciar y hacer uso de un proceso y de sus atributos.
- **function atributosAString** (): Concierte a una cadena todos los datos del proceso.

Herramientas

Tipo de programación utilizada: programación web (JavaScript).

Entorno de desarrollo: Visual Estudio Code.

Librería utilizadas: Jquery, Bootstrap , FileSaver.

Estructura del programa: HTML ,CSS