**UNIVERSIDAD TECNOLÓGICA NACIONAL**

**FACULTAD REGIONAL RESISTENCIA**

Icono

Descripción generada automáticamente

**SISTEMAS OPERATIVOS**

**Trabajo Practico Integrador:**

**Simulador de Asignación de Memoria y**

**Planificación de procesos**

Grupo Nro. 2 - Integrantes:

* Ballesteros Franco
* Bolo Luciana Agustina
* Gaczynski Mauro
* Leguizamón Sofía Violeta
* Llopi Ulises Alan

Año: 2022

Índice:

EEEEEEEEEEEEEEEEH ESTO ES UNA PEQUEÑA LISTA MIA NOMAS NO TOMAR EN SERIO…PD:el barto

1. Pequeña introducción, Problemática y planteamiento inicial
2. Funcionamiento, uso y cosas a tener en cuenta a la hora de ejecutar el código
3. Ejemplo practico con tablas del simulador
4. Conclucion

**Consigna:**

Implementar un simulador de asignación de memoria y planificación de procesos según los siguientes requerimientos.

El simulador deberá brindarla posibilidad de cargar N procesos. Para facilitar la implementación se permitirán como máximo 10 procesos y la asignación de memoria se realizará con particiones fijas. El esquema de particiones será el siguiente:

* 100K destinados al Sistema Operativo
* 250K para trabajos los más grandes.
* 120K para trabajos medianos.
* 60K para trabajos pequeños.

El programa debe permitir ingreso de nuevos procesos cuando sea posible (manteniendo en grado de multiprogramación en 5) La política de asignación de memoria será Worst-Fit, por cada proceso se debe ingresar o leer desde un archivo el Id de proceso, tamaño del proceso, tiempo de arribo y tiempo de irrupción. La planificación de CPU será dirigida por un algoritmo SJF.

El simulador deberá presentar como salida la siguiente información:

* El estado del procesador (proceso que se encuentra corriendo en ese instante)
* La tabla de particiones de memoria, la cual deberá contener (Id de partición, dirección de comienzo de partición, tamaño de la partición, id de proceso asignado a la partición, fragmentación interna)
* El estado de la cola de procesos listos.
* Listado de procesos que no se encuentran en estado de listo ni ejecución (informar el estado en que se encuentran)

**Forma de uso:**

**Ejecución:**

Para la ejecución del código no es necesario tener ninguna librería en especial, la versión de Python que se utilizo fue la 3.10.6, se recomienda usar de la versión 3.10 en adelante.

**Inicio del programa:**

Al comenzar la ejecución deberemos elegir entre dos opciones para la carga de procesos:

La opción 1 carga en la lista de nuevos unos procesos por default, que se muestran al escoger la opción en la cola de nuevos.

La opción 2 carga los procesos de un archivo

**Carga de procesos por un archivo (opción 2):**

1\_ el archivo que se puede cargar debe ser un .txt

2\_el archivo .txt tiene que tener el nombre: ArchivoDeProcesos y deberá estar ubicado en la misma carpeta que está ubicado el código .py

3\_el archivo que se desee ingresa para cargar procesos en el programa debe tener la siguiente estructura:

a\_ una primera línea que se va a saltear el código, esta línea se saltea para que quede como índices de las columnas

b\_ posterior a la primera línea, cada una de las siguientes será un proceso

c\_ los atributos de cada proceso seguirán un orden el cual se verá en el ejemplo, dato importante, el id del proceso no se deb

era ingresar, ya que se asigna automáticamente.

d\_ no tendrá que tener saltos de líneas innecesarios.

ejemplo de proceso:

tamaño ta ti

300 0 3

200 1 4

100 1 10

**Ejemplo de una simulación:**

|  |  |  |  |
| --- | --- | --- | --- |
| tabla de procesos |  |  |  |
| id | tamaño | Ta | ti |
| 0 | 170 | 0 | 2 |
| 1 | 220 | 0 | 2 |
| 2 | 140 | 1 | 3 |
| 3 | 70 | 1 | 2 |
| 4 | 100 | 1 | 4 |
| 5 | 30 | 2 | 3 |
| 6 | 30 | 2 | 5 |
| 7 | 120 | 3 | 4 |
| 8 | 60 | 4 | 5 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| t=0, mult=0 | |  | t=0, mult=2 | |  | t=1, mult=5 |
|  |  |  |  |  |  |  |
| 250kb |  |  | 80kb |  |  | 80kb |
|  |  |  |  |  |  |  |
|  |  |  | p0(170kb) |  |  | p0(170kb) |
| 120kb |  |  | 120kb |  |  | 50kb |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  | p3(70kb) |
| 60kb |  |  | 60kb |  |  | 60kb |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| 100kb |  |  | 100kb |  |  | 100kb |
| S.O. |  |  | S.O. |  |  | S.O. |
|  |  |  |  |  |  |  |
| listos=[] |  |  | listos=[p0] | |  | listos=[p0,p3] |
| ejecutando=[] | |  | ejecutando=[p0] | |  | ejecutando=[p0] |
| nuevos=[p0,p1,p2,p3,p4,p5,p6,p7,p8] | | | nuevos=[p2,p3,p4,p5,p6,p7,p8] | | | nuevos=[p5,p6,p7,p8] |
| lis/sus=[] |  |  | lis/sus=[p1] | |  | lis/sus=[p1,p2,p4] |
| terminados=[] | |  | terminados=[] | |  | terminados=[] |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| t=2, mult=5 | |  | t=4, mult=5 | |  | t=6, mult=5 |
|  |  |  |  |  |  |  |
| 30kb |  |  | 110kb |  |  | 110kb |
|  |  |  |  |  |  |  |
| p1(220kb) |  |  | p2(140kb) |  |  | p2(140kb) |
| 50kb |  |  | 50kb |  |  | 20kb |
|  |  |  |  |  |  |  |
| p3(70kb) |  |  | p3(70kb) |  |  | p4(100kb) |
| 30kb |  |  | 30kb |  |  | 30kb |
|  |  |  |  |  |  |  |
| p5(30kb) |  |  | p5(30kb) |  |  | p5(30kb) |
| 100kb |  |  | 100kb |  |  | 100kb |
| S.O. |  |  | S.O. |  |  | S.O. |
|  |  |  |  |  |  |  |
| listos=[p3,p1,p5] | |  | listos=[p2,p3,p5] | |  | listos=[p2,p4,p5] |
| ejecutando=[p1] | |  | ejecutando=[p3] | |  | ejecutando=[p2] |
| nuevos=[p6,p7,p8] | |  | nuevos=[p7,p8] | |  | nuevos=[p8] |
| lis/sus=[p2,p4] | |  | lis/sus=[p4,p6] | |  | lis/sus=[p7,p6] |
| terminados=[p0,] | |  | terminados=[p0,p1] | |  | terminados=[p0,p1,p3] |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| t=9, mult=5 | |  | t=12, mult=4 | |  | t=16, mult=3 |
|  |  |  |  |  |  |  |
| 130kb |  |  | 130kb |  |  | 130kb |
|  |  |  |  |  |  |  |
| p7(120kb) |  |  | p7(120kb) |  |  | p7(120kb) |
| 20kb |  |  | 20kb |  |  | 60kb |
|  |  |  |  |  |  |  |
| p4(100kb) |  |  | p4(100kb) |  |  | p8(60kb) |
| 30kb |  |  | 30kb |  |  | 30kb |
|  |  |  |  |  |  |  |
| p5(30kb) |  |  | p6(30kb) |  |  | p6(30kb) |
| 100kb |  |  | 100kb |  |  | 100kb |
| S.O. |  |  | S.O. |  |  | S.O. |
|  |  |  |  |  |  |  |
| listos=[p7,p4,p5] | |  | listos=[p7,p4,p6] | |  | listos=[p7,p8,p6] |
| ejecutando=[p5] | |  | ejecutando=[p4] | |  | ejecutando=[p7] |
| nuevos=[] |  |  | nuevos=[] |  |  | nuevos=[] |
| lis/sus=[p6,p8] | |  | lis/sus=[p8] | |  | lis/sus=[] |
| terminados=[p0,p1,p3,p2] | | | terminados=[p0,p1,p3,p2,p5] | | | terminados=[p0,p1,p3,p2,p5,p4] |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| t=20, mult=2 | |  | t=25, mult=1 | |  | t=30, mult=0 |
|  |  |  |  |  |  |  |
| 250kb |  |  | 250kb |  |  | 250kb |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
| 60kb |  |  | 60kb |  |  | 120kb |
|  |  |  |  |  |  |  |
| p8(60kb) |  |  | p8(60kb) |  |  |  |
| 30kb |  |  | 60kb |  |  | 60kb |
|  |  |  |  |  |  |  |
| p6(30kb) |  |  |  |  |  |  |
| 100kb |  |  | 100kb |  |  | 100kb |
| S.O. |  |  | S.O. |  |  | S.O. |
|  |  |  |  |  |  |  |
| listos=[p8,p6] | |  | listos=[p8] | |  | listos=[] |
| ejecutando=[p6] | |  | ejecutando=[p8] | |  | ejecutando=[] |
| nuevos=[] |  |  | nuevos=[] |  |  | nuevos=[] |
| lis/sus=[] |  |  | lis/sus=[] |  |  | lis/sus=[] |
| terminados=[p0,p1,p3,p2,p5,p4,p7] | | | terminados=[p0,p1,p3,p2,p5,p4,p7,p6] | | | terminados=[p0,p1,p3,p2,p5,p4,p7,p6,p8] |

Los procesos usados en el ejemplo visto están implementados en los procesos por default, por lo que podemos verificar que el código funciona de la misma manera.

**Conclusión:**