UNIVERSIDAD DE GUADALAJARA

CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS

DIVISIÓN DE ELECTRÓNICA Y COMPUTACIÓN

DEPARTAMENTO DE CIENCIAS COMPUTACIONALES

Ingeniería en computación

Seminario de Solucion de Problemas de Sistemas Operativos

SECCIÓN D-01

Actividad de Aprendizaje 2

Edgar Agustín Martinez González

CÓDIGO: 220286695

31/08/2022

**Objetivo**

Simular el procesamiento por lotes con operaciones aritmeticas basicas a las que se les designe un tiempo de ejecucion y se capturen en grupos de 3 para crear cada lote. Se deberan mostrar los procesos pendientes, en ejecucion y terminados del lote en ejecucion asi como el numero de lotes pendientes y un contador del tiempo en ejecucion de todo e lprograma.

**Desarrollo**

La simulacion del procesamiento por lotes se logro con 4 clases diferentes, la primera representa la operacion, incluye un enum para distinguir los simbolos matematicos (+, -, \*, /) y asociarlos con su operacion correspondiente. La clase operacion tambien se encarga de su representacion y conversion a partir de cadenas de texto.

pub struct Operation {

first\_operand: isize,

second\_operand: isize,

operator: Operator,

pub result: Option<isize>,

}

La siguiente clase que se usa en el programa es Process, que representa a un proceso con la informacion que se tomo del usuario y tambien dos contadores uno de tiempo restante y otro de tiempo transcurrido, que aunque son practicamente lo mismo, es mas facil actualizarlos a la par que calcular el valor de uno a partir del que estamos almacenando cada vez que necesitamos el dato.

pub struct Process {

pub id: u8,

name: String,

expected\_time: u8,

pubremaining\_time: u8,

elapsed\_time: u8,

operation: Operation,

}

La clase batch representa cada lote que puede tener hasta 3 procesos y un identificador del proceso en ejecucion. Se incluye un constructor para crear todos los lotes necesarios a partir del vector de procesos ingresado por el usuario y cada lote cuenta con sus metodos correspondientes al estado ya se para ejecutar por un segundo el lote o ver los procesos pendientes, en ejecucion o terminados.

pub struct Batch {

processes: [Option<Process>; 3],

current\_process: usize,

time\_on\_execution: u8,

}

Finalmente se mantiene una sola instancia de la clase State para pasar a la interfaz grafica el resultado del procesamiento segundo a segundo de los lotes, la informacion que contiene es el grupo de lotes a ejecutar, un identificador del lote en proceso y el contador global del programa. El estado se encarga de ejecutar los lotes en orden hasta que terminen todos y se desabiliten los cambios en el estado

pub struct State {

batches: Vec<Batch>,

current\_batch: usize,

time\_on\_execution: u8,

}

**Conclusion**

El procesamiento por lotes tenia sentido cuando se utilizaban tarjetas perforadas ya que habia un limite fisico muy facil de alcanzar de cuantas instrucciones y tarjetas podiamos metere en un momento dado en la computadora, sin embargo aun hoy en dia se le puede ver un sentido util al procesamiento por lotes, por ejemplo el agrupar procesos en lotes puede ser interesante ya no por los limites fisicos sino por motivos de organizacion que nos permitan estructurar mejores maneras de procesar datos.