

Tarea Septiembre – Diciembre 2018

Simulador de Tienda de Autoservicio

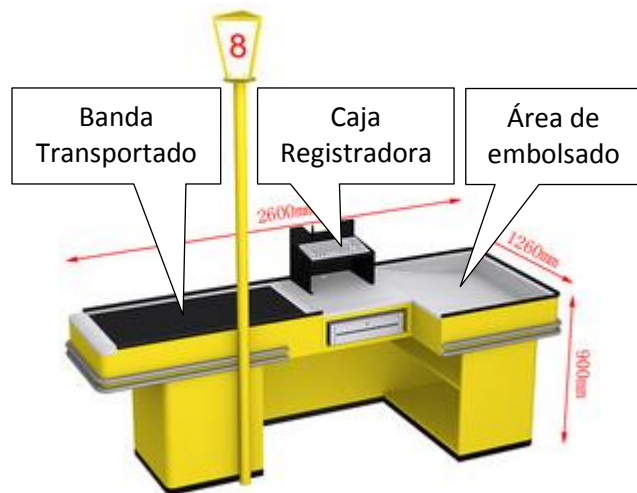
1 Introducción

Una Tienda de Autoservicio es un establecimiento en la que el cliente puede elegir y recoger personalmente las mercancías que desea comprar, generalmente de una estantería abierta. Al pagar, los clientes colocan los productos escogidos en una banda transportadora, de donde son tomados por la cajera para su totalización, y luego colocados en el área de embolsado para su respectivo embolsamiento. Sin embargo, este tipo de establecimientos sólo puede atender clientes a la velocidad de su caja. Por ello, es de primordial interés poder estimar adecuadamente la velocidad de la misma.

En este proyecto, utilizará la generación de números aleatorios para desarrollar un programa que simule una caja registradora de un supermercado. Dado el inventario de una tienda, su programa debe simular:

- Los productos en cada carrito
- La banda transportadora
- La cajera
- El embolsador

2 Características del ambiente simulado



Cada ítem en el inventario tiene un **tamaño** asociado y una **complejidad de totalización**. Los clientes de suelen colocar los ítems **más grandes** al fondo de su carrito, y los **más pequeños** arriba. Luego, al llegar al área de caja, colocan en la banda transportadora primero el ítem de más arriba (el más pequeño) y por último el de más abajo (el más grande). La cajera procesa los elementos de la banda transportadora por **orden de llegada**, y luego los **apila** en el área de embolsado. El embolsador toma los elementos en el área de embolsado desde el más accesible (cima de la pila) hasta el menos accesible (fondo de la pila).

La cajera puede comenzar a totalizar tan pronto se haya colocado el primer ítem en la banda transportadora; no tiene que esperar a que el cliente termine de pasar sus productos del carrito a la banda. Igualmente, el embolsador puede comenzar a embolsar tan pronto haya objetos en el área de embolsado; no tiene que esperar a que la cajera haya totalizado todo. Sin embargo, al terminar de atender a un cliente, la cajera debe **facturarle** a ese cliente, y no puede comenzar a cobrarle al siguiente cliente hasta que este proceso haya terminado. Podemos suponer que el cliente puede llenar la banda a capacidad instantáneamente.

El embolsador debe tomar y abrir cada bolsa antes de embolsar los ítems del área de embolsado. El embolsador debe colocar todos los ítems que quepan en la bolsa. Si hay más elementos de los que caben en una bolsa, el embolsador debe dejarlos para una segunda bolsa. Si hay menos elementos de los que caben en una bolsa, el embolsador debe cerrar la bolsa con todos estos elementos. Si un ítem es más grande que la capacidad de la bolsa, coloca aparte con los productos embolsados y se considera que ya fue procesado, pero no consume una bolsa (el embolsador no debe abrir una nueva bolsa).

3 Requerimientos del proyecto

Debe escribir un programa que simule el funcionamiento de esta actividad. El programa debe recibir un archivo que contiene el inventario de la tienda, y si no lo recibe, debe manejar el error.

Una vez iniciado, el programa presentará un menú que permitirá

1. Ejecutar la simulación
2. Cambiar las opciones de la simulación
3. Salir del programa

3.1 Funcionamiento

La simulación comienza generando los carritos que se procesarán. Cada carrito tiene un número aleatorio de elementos. Los elementos son escogidos del inventario, también de forma aleatoria, suponiendo todos los productos del inventario como igualmente probables. Puede utilizar `rand()` para generar números aleatorios entre 0 y `RAND_MAX`. Recuerde inicializar el generador de números aleatorios utilizando `srand(time(NULL))`. Los elementos se deben almacenar en una lista ordenada.

La simulación se divide en instantes de tiempo, cada uno de los cuales representa un segundo en la vida real. En cada uno de estos:

- El cliente debe cargar todos los elementos del carrito que pueda, en orden desde el más pequeño al más grande, en la banda transportadora
- La cajera debe procesar el ítem actual: si ha transcurrido el tiempo de procesamiento, lo pasa al área de embolsado. Si no, actualiza cuánto del tiempo de procesamiento ha transcurrido. Si se han procesado todos los productos del cliente, debe actualizar el tiempo de facturación.
- El embolsador debe actualizar el tiempo de procesamiento de abrir una bolsa nueva. Si ya ha transcurrido, debe armar la bolsa y colocarla con los productos embolsados

El programa tiene dos modalidades: Interactiva y Automática. En la modalidad interactiva, debe imprimirse, en cada instante de tiempo, cada una de las siguientes, en orden:

- La lista de elementos aún en el carrito
- La lista de elementos en la banda transportadora
- La lista de elementos en el área de embolsado
- La lista de elementos embolsados, colocando juntos los elementos en la misma bolsa

El programa debe esperar a que el usuario presione “enter” para pasar al siguiente instante de tiempo. Al terminar, debe imprimir cuántos instantes de tiempo tomó procesar a cada cliente, y su suma.

En la modalidad automática, sólo deben imprimirse estos últimos dos valores.

Al terminar la simulación, el programa debe agregar la corrida a un archivo de registro (log file)

3.2 Configuración de una simulación

Al entrar en las opciones de simulación, debe poder cambiar cada uno de los siguientes:

- Modalidad (Interactiva por defecto)
- Cantidad de carritos en la cola (3 por defecto)
- Cantidad máxima de productos que puede tener un carrito (30 por defecto)
- Capacidad máxima de la banda transportadora (200 cc¹ por defecto)
- Velocidad de la cajera (1,0 ops² por defecto)
- Velocidad del embolsador (4 spb³, por defecto)
- Tiempo de facturación (130 s⁴, por defecto)
- Capacidad máxima del área de embolsado (150 cc por defecto)
- Capacidad máxima de una bolsa (120 cc por defecto)

4 Formato del archivo

4.1 Archivo de entrada

En el archivo de entrada, cada línea representa un producto de la tienda. El producto se compone de:

- Nombre del producto
- Tamaño del producto en cc
- Complejidad de totalización en número de operaciones

Los valores están separados por tabuladores (el caracter de escape para un tabulador es “\t”)

Vea los archivos de ejemplo en el Moodle del curso.

4.2 Archivo de registro

En el archivo de registro, cada línea representa una corrida del simulador. Una corrida se compone de

- Nombre del archivo de entrada
- Cantidad de carritos
- Número de productos en cada carrito
- Capacidad de la banda transportadora
- Velocidad de la cajera
- Velocidad del embolsador
- Tiempo de facturación
- Capacidad del área de embolsado
- Capacidad de una bolsa
- Tiempo en procesar cada cliente
- Tiempo total (suma de los tiempos por cliente)

¹ Centímetros cúbicos

² Operaciones por segundo

³ Segundos por bolsa

⁴ Segundos

5 Evaluación

Se asignarán:

- 2 puntos por su implementación de lista ordenada
- 2 puntos por su implementación de cola
- 2 puntos por su implementación de pila
- 1 puntos por ejecutar simulaciones correctamente
- 1 punto por escribir el log correctamente
- 1 punto por dividir el programa correctamente en archivos .c y .h
- 1 punto por su *makefile*

Debe hacer uso correcto de *malloc/calloc* y *free*. El programa debe correr sin errores en las computadoras de la universidad.

6 Entrega

Debe entregar su código en un archivo comprimido (.zip, .tgz, etc.). Deberá subirlo al Moodle de la materia en la sección marcada como “📁 Tarea” hasta el viernes, 9 de febrero a las 7:30 a.m. Sólo deberá efectuar una entrega por grupo.