

Sistemas Operativos 1/2019

Laboratorio 3

Profesores:

Cristóbal Acosta (cristobal.acosta@usach.cl)

Fernando Rannou (fernando.rannou@usach.cl)

Ayudantes:

Isaac Espinoza (isaac.espinoza@usach.cl)

Marcela Rivera (marcela.rivera.c@usach.cl)

I. Objetivos Generales

Este laboratorio tiene como objetivo implementar y evaluar algoritmos de planificación de ascensores.

II. Objetivos Específicos

1. Conocer y usar las funcionalidades de `getopt()` como método de recepción de parámetros de entradas.
2. Construir funciones de lectura y escritura de archivos .csv usando `fopen()`, `fread()`, y `fwrite()`.
3. Implementar los algoritmos propuestos en el enunciado.
4. Practicar técnicas de documentación de programas.
5. Conocer y practicar uso de makefile para compilación de programas.
6. Calcular los tiempos de espera promedio para cada archivo de tráfico ingresado.
7. Proponer alguna mejora para los algoritmos propuestos.

III. Conceptos

- Edificio: Complejo de apartamentos u oficinas que tiene al menos un ascensor y uno o varios pisos. Posee un horario peak de movimiento en la mañana y en la tarde.
- Ascensor: Aparato que sirve para transportar personas o cosas de un piso a otro en un edificio. Sus direcciones de movimiento sólo son hacia arriba y hacia abajo.
- Cabina: Carro del ascensor que se encuentra dentro de un foso. Para efectos de este laboratorio, se considerará que un edificio puede tener un foso con una o varias cabinas.
- Pasajeros: Personas que son transportadas por un ascensor. Se encuentran en distintos pisos del edificio
- Llamada: Señal que recibe un elevador cuando uno o varios pasajeros quieren subirse a un ascensor. Una llamada está compuesta del segundo en el cual se realizó, la dirección de esta (hacia arriba o hacia abajo), la cantidad de pasajeros que realizan la llamada y finalmente el piso destino de esta.

IV. Enunciado

IV.A. Algoritmos de Planificación de Ascensores

Tradicionalmente, un ascensor se compone básicamente de cabinas de ascensor situadas dentro de fosos. Los carros o cabinas se encuentran conectados a cables que les permiten desplazarse verticalmente a través del foso. Generalmente, se utilizan elevadores de un sólo foso y una sola cabina, pero es común que en edificios grandes se utilicen sistemas con múltiples fosos y cabinas. Si bien la estructura puede ser distinta, el funcionamiento es similar y el fin es el mismo: Transportar personas desde un piso del edificio hacia otro. Debido a que los ascensores se utilizan en edificios que tienen varios pisos y con elevadores de un foso o múltiples fosos (por lo tanto, pueden haber varios elevadores en un solo piso) se necesita regular el comportamiento de cada ascensor y optimizar el funcionamiento de estos. Por lo tanto, es necesario utilizar algoritmos de planificación que permitan obtener el máximo provecho de cada elevador.

En este laboratorio, se estudiarán los siguientes algoritmos:

IV.A.1. *Nearest Car/Carro más cercano:*

Este algoritmo se puede considerar como la solución más básica para el problema de planificación de ascensores. Cada persona que desee utilizar el elevador, puede elegir como destino un piso superior o inferior a su piso actual, es decir, puede viajar hacia arriba o hacia abajo. Esto significa que el elevador debe conocer si el pasajero quiere dirigirse **hacia arriba** o **hacia abajo** apenas recibe una llamada. Cada llamada tiene asociada el piso desde el cual se realiza y la dirección hacia la cual el pasajero que realizó la llamada desea viajar. Por lo tanto, conociendo la dirección destino, cuando se realiza una llamada se calcula un valor conocido como: **"Figure of Suitability"** que será referido sólo como **FS**. Este valor se calculará específicamente para cada ascensor del edificio y el ascensor que tenga el **FS** más alto responderá a la llamada y recogerá a los pasajeros. Para calcular el FS, se necesitan lo siguiente:

- d: Distancia entre el piso actual del ascensor y el piso desde el cual se realizó la llamada. **$d = \text{piso actual} - \text{piso de llamada}$**
- N: Número de pisos del edificio.

Por lo tanto, el **FS** se calculará aplicando una regla (entre 4 posibles) dependiendo del estado en el cual se encuentre el ascensor al recibir la llamada. Las reglas son las siguientes:

1. Si el ascensor ya se está moviendo hacia el piso de la llamada y además la dirección de la llamada es igual a la dirección de viaje actual que tiene el ascensor. En este caso: **$FS = N + 2 - d$** .
Por ejemplo: Un ascensor está bajando del piso 8 al 2 y un pasajero realiza una llamada desde el piso 5 y su dirección destino es hacia abajo.
2. Si el ascensor ya se está moviendo hacia el piso de la llamada, pero la dirección de la llamada es distinta a la dirección de viaje actual del ascensor. En este caso: **$N + 1 - d$** .
Ejemplo: Un ascensor está bajando del piso 8 al 2 y un pasajero realiza una llamada desde el piso 5, pero su dirección destino es hacia arriba.
3. Si el ascensor se está moviendo hacia una dirección contraria al piso de la llamada, sin importar la dirección destino de la llamada. En este caso: **$FS = 1$** .
Ejemplo: Un ascensor está bajando del piso 8 al 2 y un pasajero realiza una llamada desde el piso 9, la dirección destino del pasajero no es relevante en este caso.
4. Si el ascensor está inactivo. En este caso: **$N + 1 - d$** .

Como se puede apreciar, se le da un mayor valor de FS a los ascensores que están mas cercanos al piso de la llamada y que además (si es que están en movimiento) su dirección actual es hacia el piso de la llamada e igual a la dirección destino de esta.

IV.A.2. Fixed Sectoring Common Sector System/ FSO

Este es un algoritmo especialmente diseñado para el tráfico que no es de hora punta. Básicamente, el edificio se divide en sectores (según el número de ascensores) con cantidad de pisos iguales, cada sector debe tener asignado un ascensor. Por ejemplo, un edificio de 6 pisos con 2 elevadores se divide en dos sectores.

Cada sector consiste de pisos consecutivos y el ascensor asignado a un sector debe responder a llamadas desde arriba o abajo. Un ascensor se asigna a un sector si se cumplen las siguientes condiciones:

- El ascensor no se encuentra asignado a ningún sector.
- El ascensor se encuentra en un determinado sector S.
- El sector S se encuentra vacío, es decir, no posee ningún elevador asignado.

Además, hay condiciones para que un elevador pierda su asignación actual, las cuales son las siguientes:

- Si recibe una llamada que lo fuerza a moverse fuera de su sector actual.
- Cuando el ascensor está viajando a su siguiente parada, deberá responder a las llamadas en su camino que sean hacia la misma dirección en que se está moviendo.
- Si el elevador se llena, pierde su asignación.

Finalmente, un elevador **debe** responder a las llamadas realizadas en su propio sector asignado y además a llamadas en sectores vacíos adyacentes. Los ascensores que se encuentran inactivos y sin sector asignado son situados en diferentes sectores para proveer servicio rápido a los pasajeros. Un elevador que se encuentre inactivo y con sector asignado se estaciona en su posición actual esperando por alguna llamada válida (de su propio sector o sectores adyacentes). Un elevador sin asignación se mueve al sector libre mas cercano y estaciona en él.

Se prefiere tener un elevador estacionado en el primer piso respecto a tener elevadores estacionados en cualquier otro piso, por lo tanto si el elevador que cubre el primer piso deja su sector, el ascensor estacionado más cercano debe ir al piso principal y estacionarse en él.

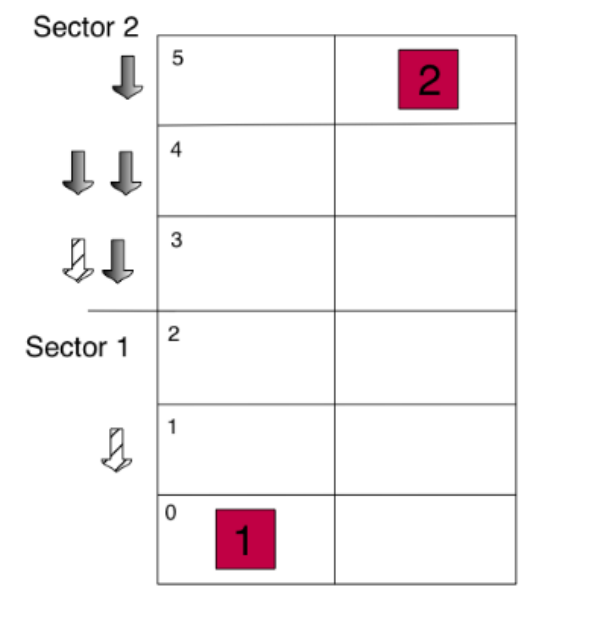


Figure 1. Ejemplo FSO, 6 pisos y 2 ascensores

IV.B. Tipos de Tráfico y Archivos de Entrada

Debido a los comportamientos que suelen observar en edificios con ascensores (por ejemplo: un complejo de oficinas, departamentos, etc), se distinguen tres tipos de tráfico que deberán ser evaluados por separado y en conjunto en este laboratorio.

IV.B.1. Tráfico de subida

Cerca del 80% de las llamadas serán realizadas desde el piso principal (primer piso) hasta un destino superior randomizado. El resto del 20% de las llamadas será completamente randomizado. Esto busca simular el tráfico durante la mañana, en el cual la gente que frecuenta un edificio llegan al primer piso y deben subir hacia su oficina de destino. Se considera entre las 7:00 hrs y las 10:00 hrs.

IV.B.2. Tráfico de bajada

Cerca del 80% de las llamadas serán realizadas desde pisos randomizados (distintos al piso terminal) que tendrán como destino el piso principal. El 20% restante será completamente randomizado. Esto busca simular el tráfico de la tarde, en el cual un grupo de trabajadores finaliza su horario laboral y obligatoriamente para dejar el edificio debe bajar al piso principal. Se considera entre las 17:00 hrs y las 20:00 hrs.

IV.B.3. Tráfico Ordinario

Piso de llamada y destino randomizado. Busca simular el tiempo entre los horarios peak de subida (mañana) y peak de bajada (tarde). Se considera entre las 10:00 hrs y las 17:00 hrs.

IV.C. Archivos de Entrada

Por lo tanto, se tendrán tres archivos de entrada con simulaciones de los tráficos ya descritos. El formato de los archivos será el siguiente:

Ejemplo:

1,1,5,U,2
45,1,7,U,3
60,2,1,D,1

Los archivos tienen formato **.csv** y la separación es por comas. Cada línea del archivo consiste en una **llamada** realizada por un pasajero o por un grupo de pasajeros. El primer elemento de la línea consiste en la hora en que fue realizada la llamada. Los siguientes elementos consisten en: piso inicial, piso destino, dirección de la llamada y número de pasajeros que realizaron la llamada. Por lo tanto, la primera llamada de ejemplo tiene los siguientes valores:

- Tiempo/Segundo en que se realizó la llamada: Segundo 1
- Piso desde el que se realizó la llamada: 1
- Piso destino: 5
- Dirección del viaje: U (Up/Arriba)
- Número de pasajeros que realizaron esta llamada: 2

Cabe destacar que los 3 archivos tienen el mismo formato, solo cambia la distribución de las llamadas y el rango horario según lo mencionado en la sección anterior.

IV.D. Medición del Rendimiento

Para medir la eficiencia de los algoritmos propuestos, se necesita construir un simulador que pueda simular los tráficos propuestos de un ascensor en un edificio, debido a que no es posible probar nuevos algoritmos en ascensores reales. Además, la mayoría de los simuladores existentes son pagados.

El rendimiento será evaluado con el tiempo de espera promedio desde que un pasajero realiza una llamada, hasta que es recogido por un ascensor. En total se obtendrán 4 tiempos promedio:

- Tiempo promedio de espera para tráfico de subida.
- Tiempo promedio de espera para tráfico de bajada.
- Tiempo promedio de espera para tráfico ordinario.
- Tiempo promedio de espera general.

IV.E. Parámetros constantes de un ascensor

Para poder medir el tiempo de espera promedio, se necesitan utilizar valores constantes para cada acción del elevador y así poder simular correctamente un edificio con varios ascensores. A continuación se listan las acciones de un elevador con sus respectivos tiempos:

- Tiempo que demora en avanzar un piso: 1 segundo, se considera que el ascensor está avanzando a velocidad máxima.
- Tiempo de carga de un pasajero: 1 segundo por pasajero.
- Tiempo en abrir puertas: 1 segundo.
- Capacidad de carga: Máximo 20 pasajeros por elevador.
- Supuesto 1: Si un grupo de X personas realiza una llamada, todas tienen la misma dirección destino y deben subirse juntas a un elevador. En caso de que ningún elevador pueda atenderlas (debido a la capacidad), esperarán hasta que un ascensor pueda recoger al grupo completo.
- Supuesto 2: Se prioriza atender las llamadas hasta que el elevador se llene. Si al atender una llamada se pasa por un piso "destino", se pueden descargar pasajeros. Sino, se empezarán a atender los destinos una vez el elevador esté lleno y no se puedan atender más llamadas o una vez se hayan atendido todas las llamadas del ascensor.
- Supuesto 3: Si un ascensor está atendiendo una llamada y se encuentra viajando en un piso "X", y en ese momento llega una llamada nueva desde el mismo piso, entonces si se cumple el criterio del algoritmo el ascensor puede atender inmediatamente esa llamada.
- Supuesto 4:

Por ejemplo, si un elevador está inactivo en el piso 1, recibe una llamada desde el piso 2, se sube un sólo pasajero y su destino es el piso 9:

- Paso 1: El elevador debe ascender desde el piso 1 hasta el piso 2. Demora 1 segundo.
- Paso 2: Ya en el piso 1, el elevador debe abrir sus puertas. Demora 1 segundo.
- Paso 3: El recoger al pasajero. Demora 1 segundo.
- Paso 4: Como el elevador no posee más llamadas asignadas que atender, puede atender inmediatamente el destino y por lo tanto sube desde el piso 2 hasta el piso 9 (destino del pasajero que recién abordó el ascensor). Demora 7 segundos.
- Paso 5: Ya en el piso 9, el elevador debe abrir sus puertas. Demora 1 segundo.
- Paso 6: El elevador descarga al pasajero. Demora 1 segundo.

Por lo tanto:

Tiempo Total: 12 segundos

Tiempo de Espera: Sólo se considera desde que se realizó la llamada hasta que se llegó al piso del pasajero, por lo tanto el tiempo de espera fue de 1 segundo.

IV.F. Simulador

Para el diseño del simulador, se deben considerar los supuestos ya mencionados respecto a la descarga de pasajeros y a la resolución de llamadas.

El simulador debe ser implementado a base de "ticks" que representarán el paso de un segundo. Se debe simular el paso del tiempo en segundos $[0, n]$ y en cada segundo se debe determinar si:

- Existe una llamada nueva en el segundo actual.
- En caso de existir, se debe asignar inmediatamente algún elevador a dicha llamada nueva.

- Para cada elevador, se debe determinar un tipo de acción que podrá ser: moverse un piso hacia arriba o abajo para atender una llamada o un destino, descargar un pasajero, mantenerse inactivo, reposicionarse (en caso de algoritmo FSO).
- Actualizar los tiempos de espera de cada llamada y todas las métricas que deben ser medidas respecto a cada elevador.

Además, se deberá modelar cada elevador como una estructura con parámetros que faciliten el manejo de eventos (arreglo, lista o cola con llamadas asignadas y destinos de dichas llamadas, número actual de pasajeros a bordo, etc) y el registro de las métricas que deben medirse (pisos que el elevador se ha movido, tiempo en movimiento, etc).

V. Salida del programa

Deberá implementar un programa que simule el uso de N ascensores, para M pasajeros, haciendo uso de los algoritmos propuestos previamente. Finalmente debe escribir un archivo de texto para cada algoritmo, el cuál debe contener:

1. Tiempo promedio de espera para tráfico de subida para cada ascensor.
2. Tiempo promedio de espera para tráfico de bajada para cada ascensor.
3. Tiempo promedio de espera para tráfico ordinario para cada ascensor.
4. Tiempo promedio de espera general de cada ascensor.
5. Total de pisos recorridos por cada ascensor.
6. Tiempo total de funcionamiento de cada ascensor.
7. Proponer una mejora en alguno de los algoritmos utilizados para disminuir el tiempo de espera de cada pasajero.

El programa se ejecutará usando los siguientes argumentos (ejemplo):

```
$ ./lab3 -u subida.csv -d bajada.csv -o ordinario.csv -n npisos -e nascensores -b
```

- **-u:** archivo de tráfico de subida
- **-d:** archivo de tráfico de bajada
- **-o:** archivo de tráfico ordinario
- **-n:** número de pisos
- **-e:** número de ascensores
- **-b:** bandera o flag que permite indicar si se quiere ver el tiempo total y el tiempo de espera para cada archivo.

Ejemplo de compilación y ejecución:

```
>> make
>> ./lab3 -u subida.csv -d bajada.csv -o random.csv -n 30 -e 2 -b
```

Finalmente, la salida por consola debiese ser:

```
Tiempo total ascensor 1 NC: 1000 s.
Tiempo total ascensor 2 NC: 1500 s.
Total de pisos recorridos ascensor 1 NC: 50.
Total de pisos recorridos ascensor 2 NC: 70.
Tiempo total ascensor 1 FSO: 1200 s.
Tiempo total ascensor 2 FSO: 1800 s.
Total de pisos recorridos ascensor 1 FSO: 65.
Total de pisos recorridos ascensor 2 FSO: 98.
```

Al ejecutar sin el flag -b, funciona igual al anterior siendo la única diferencia que no se deberá mostrar ningún tipo de salida por consola.

Respecto a los archivos de salida, deben tener el siguiente formato:

Nearest Car:

Tiempo promedio espera Subida: 20.45 segundos.
Tiempo promedio espera Bajada: 14.65 segundos.
Tiempo promedio espera Ordinario: 43.32 segundos.
Tiempo promedio espera General: 65.44 segundos.
Tiempo total ascensor 1: 1000 s.
Tiempo total ascensor 2: 1500 s.
Total de pisos recorridos ascensor 1: 50.
Total de pisos recorridos ascensor 2: 70.

FS0:

Tiempo promedio espera Subida: 24.45 segundos.
Tiempo promedio espera Bajada: 53.65 segundos.
Tiempo promedio espera Ordinario: 53.32 segundos.
Tiempo promedio espera General: 75.44 segundos.
Tiempo total ascensor 1: 1200 s.
Tiempo total ascensor 2: 1800 s.
Total de pisos recorridos ascensor 1: 65.
Total de pisos recorridos ascensor 2: 98.

Como requerimientos no funcionales, se exige lo siguiente:

- Debe funcionar en sistemas operativos con kernel Linux.
- Debe ser implementado en lenguaje de programación C.
- Se debe utilizar un archivo Makefile para su compilación.
- Realizar el programa utilizando buenas prácticas, dado que este laboratorio no contiene manual de usuario ni informe, es necesario que todo esté debidamente comentado.
- Que el programa principal esté desacoplado, es decir, que se desarrollen las funciones correspondientes en otro archivo .c para mayor entendimiento de la ejecución.

VI. Entregables

El laboratorio es en parejas y se descontará 1 punto por día de atraso. Debe subir en un archivo comprimido a usachvirtual los siguientes entregables:

- **Makefile:** Archivo para make que compila el programa.
- **archivos .c y .h** Se debe tener como mínimo un archivo .c principal con el main del programa. Se debe tener mínimo un archivo .h que tenga cabeceras de funciones, estructuras o datos globales. Se deben comentar todas las funciones de la forma:

```
//Entradas: explicar que se recibe  
//Funcionamiento: explicar que hace  
//Salidas: explicar que se retorna
```

- Trabajos con códigos que hayan sido copiados de un trabajo de otro grupo serán calificados con la nota mínima.

El archivo comprimido debe llamarse: RUTESTUDIANTE1_RUTESTUDIANTE2.zip

Ejemplo: 19689333k_186593220.zip

NOTA: los laboratorios son en parejas, las cuales deben ser de la misma sección. De lo contrario no se revisarán laboratorios.

VII. Fecha de entrega

Domingo 14 de Julio.