Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación
IIC2333 Sistemas Operativos y Redes

Tarea 1

1. Problemas Individuales

Estos ejercicios tienen como objetivo que usted repase los conceptos vistos en clase.

1.1. Problema 1

Dado el siguiente set de procesos:

Proceso	T. Llegada	T. Ejecución	Prioridad
1	0	3	2
2	1	5	4
3	2	1	0
4	4	7	1
5	7	3	6
6	11	2	5
7	15	4	3
8	16	2	7

a) Realice los diagrama de Gantt para los siguientes algoritmos de planificación: FCFS, SJF, Prioridad con expropiación (siendo 0 la mayor prioridad).
b) Calcular el throughput, turnaround, tiempo de espera, tiempo de respuesta para cada uno de los procesos y para cada uno de los algoritmos de la parte a.
c) Explique cuales son las ventajas y desventajas de estos algoritmos. Justifique.

1.2. Problema 2

Si sumamos el cuadrado de los cinco primeros numeros naturales obtenemos:

$$1^2 + 2^2 + ... + 5^2 = 55 \tag{1}$$

Mientras que el cuadrado de la suma de los cinco primeros numeros naturales es:

$$(1 + 2 + ... + 5)^2 = 152 = 225 \quad (2)$$

La diferencia, en este caso, entre (2) y (1) es 225 - 55 = 170. Escriba un programa en Python que reciba un numero y calcule la diferencia entre el caso (2) y (1).

1.3. Problema 3

a) Hay cinco filósofos sentados alrededor de una mesa que pasan su vida cenando y pensando. Cada uno dispone de un plato de arroz y un palillo a la izquierda de su plato, pero para comer son necesarios dos palillos y cada filósofo sólo puede coger el que está a su izquierda o el que hay a su derecha. Con un solo palillo en la mano no tienen más remedio que esperar hasta que atrapen otro y puedan seguir comiendo. Si dos filósofos adyacentes intentan tomar el mismo palillo a la vez se produce una condición de carrera: ambos compiten por lo mismo pero uno se queda sin comer. Describa los criterios de exclusión mutua, sincronización y deadlock usando analogías del problema descrito anteriormente.

b) Sean dos threads, A y B, que compiten entre ellos. A intenta incrementar un contador común mientras que B intenta disminuir el contador.

```
Thread A
i = 0;
while(i < 10)
    i = i + 1;
printf("Gana A");
```

```
Thread B
i = 0;
while(i > - 10)
    i = i - 1;
printf("Gana B");
```

1) Quien gana?
2) Está garantizado que alguien gane? Si? No? Porqué?
3) Si ambos threads tuviesen sus propias CPU corriendo a la misma velocidad? Es posible asegurar que se ejecuten para siempre?

2. Problemas Grupales

En esta parte de la tarea usted debe implementar el sistema operativo de un celular. Específicamente debe implementar un programa en Python que simule un sistema operativo multiprogramable, el cual permita realizar los siguientes procesos básicos:

* Hacer/recibir llamadas. Guardar en el historial la hora en que fue efectuada/recibida la llamada y el tiempo de duración.

* Enviar/recibir mensajes de texto. Guardar los mensajes recibidos en memoria.

* Revisar la agenda de contactos y realizar llamadas desde ella.

* Revisar historial de llamadas y de mensajes de texto.

Para ello deberá implementar las clases básicas necesarias para emular todos los objetos que pueda necesitar. Por otro parte deberá implementar un algoritmo de agendamiento de Prioridad.

Además de lo anterior debe existir un proceso que permita ver los procesos que están corriendo en ese momento (tome como referencia la función top de Linux). Y también debe existir un proceso que permita ejecutar procesos indicando el nombre (Utilice el formato definido más adelante).

La memoria del celular debe simularla con archivos de texto donde se guarde y lea toda la información relevante a los procesos que se ejecutan.

Tenga en cuenta que como todo sistema operativo debe manejar las interrupciones que se puedan dar entre los procesos. Por ejemplo, si mientras está escribiendo un mensaje de texto entra una llamada, no se puede perder lo que se estaba escribiendo por contestar.

Finalmente, para poder probar la tarea, su programa debe aceptar como input un archivo de texto que contenga la lista de procesos que se van a ejecutar. En cada línea del archivo debe ir un proceso, indicando:

Nombre_Proceso, Fecha_Ejecucion, Tipo_Proceso, Prioridad_Base, [Opciones]

Donde [Opciones] dependerá del tipo de proceso:

- Si se trata de hacer una llamada Tipo_Proceso valdrá 1 y en opciones iría el numero y el tiempo de ejecución separados por un punto coma.

- Si se trata de recibir una llamada Tipo_Proceso valdrá 2 y en opciones iría el numero y el tiempo de ejecución separados por un punto coma.

- Si se trata de enviar mensajes de texto Tipo_Proceso valdrá 3 y en opciones iría el receptor y el texto que se quiere enviar separados por un punto coma. **El tiempo de envío de un mensaje es en segundos y se calcula como 20 ms por letra, redondee hacia arriba. Note que esto es enviar el mensaje y no escribir el mensaje.**

- Si se trata de recibir mensajes de texto Tipo_Proceso valdrá 4 y en opciones iría el emisor y el texto que se debe recibir separados por un punto coma. **El tiempo de envío de un mensaje es en segundos y se calcula como 20 ms por letra, redondee hacia arriba.**
- Si se trata de agregar un Contacto Tipo_Proceso valdrá 5 y en opciones iría el nombre y el número telefónico del contacto separados por un punto coma.
- Si se trata de proceso cualquiera Tipo_Proceso valdrá 6 y en opciones iría el tiempo de ejecución.
- Si se trata de mandar la ubicación Tipo_Proceso valdrá 7 y el tiempo de ejecución será de 2 segundos
- Si se trata de ver la ubicación Tipo_Proceso valdrá 8 y en opciones iría el tiempo de ejecución.
- Si se trata de jugar Tipo_Proceso valdrá 9 y en opciones iría el tiempo de ejecución.
- Si se trata de escuchar música Tipo_Proceso valdrá 10 y en opciones iría el tiempo de ejecución.

Fecha de Entrega: Martes 16 de Abril hasta las 23:59:59.