



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
1	0	8
2	1	4
3	4	6
4	6	5
5	7	3
6	11	2

- a) Cree las cartas Gantt de ejecución de estos procesos utilizando los siguientes algoritmos de agendamiento: FCFS, SJF con expropiación y RR(4).
- b) Calcular el throughput, turnaround, tiempo de espera y tiempo de respuesta para cada uno de los procesos y para cada uno de los algoritmos de la parte a).
- c) ¿Qué algoritmo de agendamiento convendría usar para cada uno de los siguientes casos? Explique:
- Los procesos llegan entre sí a intervalos largos de tiempo
 - La eficiencia del sistema está medida por el porcentaje de procesos completados
 - Todos los procesos tienen el mismo tiempo de ejecución

1.2. Problema 2

Considerando la clase Botella escrita más abajo y una lista de 40 Botellas. Escriba un programa en python que simule 3 maquinas que recorren la lista, estas deben tapar y llenar todas las botellas simultáneamente. Su programa no debe lanzar ninguna excepción, y todas las botellas deben terminar tapadas y llenas. Es decir, todas las botellas de la lista deben retorna True al invocar el método “estaLista”. Para resolver el problema use sus conocimientos de Hilos y Sincronización de procesos.

Nota: Puede agregar líneas de código a la clase Botella. Pero si su solución es “truculenta”, no se le dará ningún punto en esta pregunta.

```
class Botella:
    __tapa=False
    __llena=False
    def tapar(self):
        if(not self.__tapa):
            self.__tapa=True
        else:
            raise Exception("Intentaste tapar dos veces la misma botella")
    def llenar(self):
        if(not self.__llena and not self.__tapa):
            self.__llena=True
        else:
            raise Exception("La botella se ha rebalsado")

    def estaVacía(self):
        return not self.__llena
    def estaTapada(self):
        return self.__tapa
    def estaLista(self):
        return ((not self.estaVacía()) and self.estaTapada())
```

1.3. Problema 3

Para la siguiente situación, deberán escribir un pseudo-código utilizando los conceptos de sincronización vistos en clases. Pueden entregar una respuesta para la parte a y b o dos por separado.

a) El viejito pascuero, duerme durante el año para recargar energías para el gran día cuando debe recorrer el mundo entero repartiendo regalos a los niños. Durante este tiempo, los ayudantes de santa se dedican a hacer estos regalos. Sin embargo, algunas veces tienen problemas o consultas respecto a los juguetes. Para que no despertaran al viejito cada vez, éste decidió establecer la regla de que lo despertaran cuando 3 de ellos necesitaran consultarlo y sólo atendería 3 a la vez. Lo anterior quiere decir que, mientras tres están siendo ayudados, el cuarto debe esperar a que uno esté listo y puede ir a atenderse.

b) Además de los elfos, están los renos. Estos también se toman vacaciones durante el año, cada uno por separado y vuelven un día antes de la navidad. Cuando el séptimo reno llega, debe despertar a Santa para que prepare el trineo. Si el séptimo reno lo despierta y justo hay elfos con preguntas, estos deberán esperar a la vuelta de navidad para preguntar.

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 Colas de multinivel donde las llamadas son las mas prioritarias, luego los mensajes y por ultimo los otros procesos. Recuerde que estos últimos pueden correr en paralelo.

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: Lunes 9 de Septiembre hasta las 23:59:59.