



PONTIFICIA UNIVERSIDAD JAVERIANA - CALI
FACULTAD DE INGENIERIA
SISTEMAS OPERATIVOS
Arquitectura de Software
2020.2
— Tercer Parcial —

Nombre: _____

"El estudiante de la Pontificia Universidad Javeriana, como agente de su propia formación, es responsable de la Identidad Institucional, uno de cuyos cimientos es tener como habito un comportamiento ético en todos los ámbitos de la vida. En este sentido me comprometo a realizar con total integridad esta evaluación, solamente empleando los recursos autorizados para su desarrollo."

Consejo Académico, Acta Nro 79, abril 19 de 2004

1. (4 puntos) Realizar un programa que permita la gestion de una región contigua de memoria de tamaño MAX donde las direcciones pueden ir desde 0 ... MAX - 1. Su programa debe responder a cuatro peticiones diferentes:
 - (a) Solicitud de un bloque contiguo de memoria
 - (b) Liberación de un bloque contiguo de memoria
 - (c) Compactación de huecos de memoria no utilizados en un solo bloque
 - (d) Informar de las regiones de memoria libre y asignada

Su programa recibirá la cantidad inicial de memoria al arrancar. Para ejemplo, lo siguiente inicializa el programa con 1 MB (1.048.576 bytes) de memoria:

```
./allocator 1048576
```

Una vez que su programa haya arrancado, presentará al usuario el siguiente prompt:

```
allocator>
```

Luego responderá a los siguientes comandos:

- (a) RQ (solicitud)
- (b) RL (liberación)
- (c) C (compactar)
- (d) STAT (informe de estado)
- (e) X (salir)

Una solicitud de 40.000 bytes aparecerá de la siguiente manera:

```
allocator>RQ P0 40000 W
```

El primer parámetro del comando RQ es el nuevo proceso que requiere la memoria, seguido de la cantidad de memoria que se solicita, y finalmente la estrategia. (En esta situación, "W" se refiere al peor ajuste). De manera similar, una liberación aparecerá como:

```
allocator>RL P0
```

Este comando liberará la memoria que ha sido asignada al proceso P0. La orden de compactación se introduce como:

```
allocator>C
```

Este comando compactará los huecos de memoria no utilizados en una región.

Por último, el comando STAT para informar del estado de la memoria se introduce como:

allocator>STAT

Con este comando, su programa informará de las regiones de memoria que están asignadas y las regiones no utilizadas. Por ejemplo, una posible organización de la asignación de memoria sería la siguiente de la asignación de memoria sería la siguiente:

```
Addresses [0:315000] Proceso P1
Addresses [315001: 512500] Proceso P3
Addresses [512501:625575] No utilizado
Addresses [625575:725100] Proceso P6
Addresses [725001] . . .
```

Asignación de memoria Su programa asignará la memoria utilizando uno de los tres enfoques vistos en clase, dependiendo de la bandera que se pase al comando RQ. RQ. Las banderas son:

- F -primer ajuste
- B -mejor ajuste
- W - peor ajuste

Esto requerirá que su programa lleve la cuenta de los diferentes agujeros que representan la memoria disponible. Cuando llegue una solicitud de memoria, asignará la memoria de uno de los agujeros disponibles basándose en la estrategia de asignación.

Si no hay suficiente memoria para asignar a una solicitud, emitirá un mensaje de error y rechazará la solicitud. y rechazará la solicitud.

Su programa también necesitará llevar la cuenta de qué región de memoria ha sido asignada a cada proceso. Esto es necesario para soportar el comando STAT y también es necesario cuando se libera memoria mediante el comando RL, ya que el proceso que libera la memoria se pasa a este comando. Si una partición que se libera liberada es adyacente a un agujero existente, asegúrese de combinar los dos agujeros en un único agujero.

Compactación Si el usuario introduce el comando C, su programa compactará el conjunto de agujeros en un solo agujero más grande. Por ejemplo, si tiene cuatro agujeros separados de tamaño 550KB, 375KB, 1.900KB, y 4.500KB , su programa combinará estos cuatro agujeros en un único agujero de 7.325 KB.

2. (1 punto) MooseFS es un sistema de archivos distribuidos, tolerante a fallos, rapido, escalable y de alta velocidad. En este punto usted deberá crear un cluster de MooseFS usando docker, para esto siga las instrucciones de esta pagina: https://medium.com/@karol_majek/moosefs-multinode-cluster-using-docker-512d0619360a que le permita crear un archivo en el cluster y entrar a la pagina de administración de MooseFS. Adicional explique brevemente las características que ofrece MooseFS.

Notas generales:

- Puede ser trabajado en parejas.
- Realizar un video donde sustenten el parcial explicando linea a linea los códigos.
- En el video se debe ver todo el tiempo el rostro de cada uno de los autores.