



Sistemas Operativos
Segundo Cuatrimestre 2022
Trabajo Práctico N°2
Grupo 14

Integrantes:

- Gazzaneo, Lautaro Nicolas
- Limachi, Desiree Melisa

Decisiones Tomadas:

Para el custom memory manager se usó una lista con bloques ordenada por el tamaño del bloque así que al hacer malloc se va a buscar por la lista hasta encontrar el primer bloque con igual tamaño requerido o terminar la lista, si el bloque es muy grande se puede dividir a la mitad y se genera el nuevo bloque seguido del anterior, entonces allí quitará al bloque de la lista y cuando se lo libere se lo volverá a poner en ella.

También se podrá insertar bloques en la lista en la cual va buscando hasta que encuentra un bloque de tamaño igual o mayor o el final y a inserta el bloque en la lista.

Para el scheduler se decidió usar una lista circular con los procesos y que cada nuevo proceso se encola al final de la lista, también se definió la prioridad de 0 a 19 con la prioridad de default siendo el promedio, en este caso 10. Se definieron enums con los estados y modos de los procesos (Ready/Blocked y Back/Fore). Cuando se crea el primer proceso que lo creamos nosotros este solo espera para que un proceso "real" llegue para matarse a sí mismo. La fórmula para calcular tickets y tiempo de ejecución están comentadas en el scheduler.h

Instrucciones de compilación y ejecución:

Primero que todo se debería entrar al docker con la imagen dada por la cátedra y dentro de la terminal de docker se deberían realizar los siguientes comandos:

Primero al estar en la carpeta del proyecto se necesitaría hacer "cd ./Toolchain" y luego "make all" para hacerle el build de la Toolchain luego volvemos con "cd .." al directorio principal del proyecto y hacemos "make all".

Luego se saldría de la terminal de docker con un "exit". Para luego poder ejecutar el proyecto con "./run.sh". Podría darse un error con las imágenes para eso se debería ejecutar del directorio principal del proyecto "sudo chmod 777 ./Images/*"

Si se quiere usar el sistema Buddy de memoria antes de hacer el make all en el directorio principal se tiene que ir a la carpeta kernel con "cd ./Kernel" y luego hacer "make MM=BUDDY all" y volver al directorio principal después con "cd .." y seguir lo mismo desde ahí.

Demostración de los requerimientos:

Principalmente existen los comandos de test para realizar los testing de los requerimientos especificados, también en la shell para las aplicaciones en userspace con su comando respectivo se podrá mostrar su funcionamiento.

Limitaciones y Problemas durante el desarrollo:

A la mitad del desarrollo el grupo se separó debido a diferencias quedando solo los dos integrantes actuales para realizar la mayoría del proyecto casi desde el inicio.

Se tuvo problemas al implementar el funcionamiento en especial con Memory Management a utilizar, se fue resolviendo el problema con distintos cambios en la implementación ya que veíamos fallas en los tests.

Durante la implementación de Phylo con respecto a agregar o borrar, encontramos un problema con los semáforos y no se logró finalizar el proceso correctamente, si lo inicializamos para que se tome en cuenta algunos filósofos al inicio.