



Instituto Tecnológico de Buenos Aires

Sistemas Operativos

Informe 2: Construcción del Núcleo de un Sistema Operativo y estructuras de administración de recursos.

Integrantes grupo 5:

Perez Riviera, Mateo 61170

Szejer, Ian 61171

Ortu, Agustina Sol 61548

Profesores:

Aquili, Alejo Ezequiel

Mogni, Guido Matias

Godio, Ariel

Decisiones tomadas durante el desarrollo

Se discutió bastante en el equipo como iban a ser las estructuras que manejaría el scheduler. Terminamos eligiendo usar listas de listas, donde el scheduler tiene un puntero a la primera lista la cual tendrá la más baja prioridad, luego, las demás están conectadas en orden creciente de prioridad. Además, cada lista tiene una lista de punteros a procesos propia, donde cada proceso tiene la prioridad correspondiente a la lista en la que se encuentra y se ordenan según orden de llegada a la lista.

Instrucciones de compilación y linkedicion

Para compilar y linkeditar todo el proyecto, se debe tener la imagen de docker agodio/itba-so:1.0 y desde docker realizar el make deseado.

Al momento de realizar el make hay dos opciones:

1. *make all* -> corre el programa con el memory manager creado por el grupo
2. *make MEM_MANAGER=BUDDY* -> corre el programa con el buddy creado por el grupo

Es importante tener en cuenta que es necesario correr el comando *make clean* antes de cada creación para evitar conflictos por archivos residuales.

Finalmente para ejecutarlo se debe correr el comando *./run.sh*

Limitaciones

No llegamos a realizar todos los testeos deseados, a pesar de que consideramos que todo tiene una lógica, no estamos seguros de su correcta funcionalidad.

Hay algunos warnings que no llegamos a corregir.

Los pipes no pudieron ser implementados correctamente.

Problemas encontrados y su solución

Nuestro principal problema que nos retrasó mucho tiempo fue que nuestro programa crasheaba al intentar cargar las interrupciones. Luego de mucho testeo y una reunión con Alejo, llegamos a la conclusión que había algún problema en la compilación o ejecución de nuestro programa ya que fallaba en nuestras máquinas pero no en la de Alejo. Para resolverlo realizamos la compilación desde docker

Al hacer el guardado del contexto se guarda la posición donde estan cargados los argumentos pero al recuperarla los valores no son los mismos, esto no lo pudimos resolver.

Citas de fragmentos de código obtenidos de otras fuentes

El buddy memory manager se basó en el archivo .c del siguiente repositorio gitHub: <https://github.com/evanw/buddy-malloc>

https://github.com/IanSzejer/SO_TP2

Hash: 7b07fd610ea9b7f41b7354d1cc56c6fefa94a4fa