



Trabajo Práctico 2: Tercera Entrega

72.11 - Sistemas Operativos

Segundo Cuatrimestre 2024

Integrantes del **Grupo 21**:

62872	Matías LEPORINI
63382	Camila LEE
63074	Ana NEGRE

Instrucciones de Compilación

Para crear el contenedor dentro del directorio que contiene el repositorio:

```
docker pull agodio/itba-so-multi-platform:3.0

cd $MyDir

docker run -v "${PWD}:/root" --security-opt seccomp:unconfined -ti
agodio/itba-so-multi-platform:3.0
```

Una vez iniciado el contenedor, correr los comandos:

```
cd root

cd TP2-SistemasOperativos-72.11
```

Se provee para la compilación un script que compila el proyecto en su totalidad. El mismo compila tanto el kernel como el módulo de usuario y la toolchain. Para utilizarlo, dentro del contenedor puede correr el comando:

```
bash ./compile.sh
```

Instrucciones de Ejecución

Para ejecutar el proyecto, puede hacer uso del script provisto para tal fin. Fuera del contenedor, debe ejecutar:

```
bash ./run.sh
```

Decisiones de Desarrollo

Para esta entrega parcial se implementó la posibilidad del uso de semáforos a través de syscalls. Para la implementación se utilizaron las funciones proporcionadas por la cátedra para la obtención del acceso exclusivo a la zona crítica. Además, se decidió contar con un *lock* global para impedir *race-conditions* entre procesos al intentar modificar variables de los semáforos, y un *lock* individual para cada semáforo, el cual es el mutex “en sí”.

Paralelamente, se implementó la lógica necesaria para establecer la “familia” de procesos. Anteriormente, cada proceso era independiente, por lo que no existía la

posibilidad de esperar a procesos hijos, crear un hijo, etc. Esto se implementó (no solo a raíz de los requerimientos funcionales del trabajo práctico), sino también para poder ejecutar el test de sincronización.

Observaciones

Se aclara que, dada la limitación de la terminal actual (no acepta argumentos), se decidió establecer por el momento como constante la cantidad de iteraciones del test, la cual puede modificarse y recompilar el sistema. Sin embargo, sí se incluyó la posibilidad de probar correr el test sin el uso de semáforos.

Fragmentos de Código Reutilizado

En esta entrega, la implementación de las funciones de semáforos se basó en los conceptos discutidos en las clases prácticas y teóricas.