

Informe de Trabajo Práctico N° 2 – Kernel

Grupo 20 – Alfieri, Di Candia, Díaz Varela

Introducción

Este informe destaca únicamente aspectos a tener en cuenta durante la corrección que no sean evidentes al ejecutar el TP.

Instrucciones de Compilación y Ejecución

Para compilar y correr el proyecto, utilizar los siguientes comandos desde la raíz del proyecto:

```
cd Toolchain
```

```
make all
```

```
cd ..
```

```
./run.sh
```

Al compilar el proyecto se usa como predeterminado el Memory Manager básico (al que referenciamos como **OUR** a lo largo del proyecto).

Para compilar el proyecto con el Buddy Memory Manager, utilizar el siguiente comando:

```
./run.sh buddy
```

En caso de que no funcione el script:

```
sudo make buddy
```

```
sudo make our
```

```
sudo qemu-system-x86_64 -hda Image/x64BareBonesImage.qcow2 -m 512
```

Decisiones de Diseño

- **Implementación de Semáforos:**
Cada semáforo tiene su propio spinlock para evitar condiciones de carrera y se utiliza una lista de espera para los procesos bloqueados. Se decidió barrer procesos muertos al reanudar para evitar *memory leaks* en la cola de espera.
- **Asignación de memoria:**
Se implementaron dos estrategias de gestión de memoria: Buddy System (**BUDDY**) y un heap simple con bloques enlazados (**OUR**). La selección se realiza mediante macros de compilación.

Instrucciones para demostrar el funcionamiento

Ingresando el comando:

help

en la shell, se despliega la lista de comandos para el usuario a través de los cuales se demuestran los funcionamientos requeridos.

Limitaciones

1. **Fragmentación Interna**
 - El gestor de memoria simple (**OUR**) puede sufrir de fragmentación interna.
2. **Ausencia de Protección de Memoria**
 - Actualmente, el sistema no implementa un mecanismo de protección de memoria entre procesos. Esto significa que un proceso malicioso o con errores podría potencialmente corromper el espacio de memoria de otro proceso si intenta acceder fuera de sus límites asignados. Esto se debe a que todos los procesos comparten un espacio de memoria.
3. **Alcance de Pipes**
 - La implementación actual de *pipes* no soporta la conexión de más de dos procesos (**p1 | p2 | p3**), limitándose a un encadenamiento simple (**p1 | p2**).
4. **Cantidades Máximas**
 - Como por ejemplo la limitación definida a la cantidad de semáforos por **MAX_SEMAPHORES**.
 - limitación de cantidad máxima de procesos a crear, definida como **MAX_PROCESSES**.

Consideramos que cualquier aspecto no mencionado en este informe es evidente a partir de la lectura del código fuente en sí.

