Universidad de San Carlos de Guatemala

Faculta de Ingeniería

Sistemas Operativos 1

Inti Andrés Samayoa Ortíz

201504002

Módulos del Kernel

Módulo de Memoria (sysinfo)

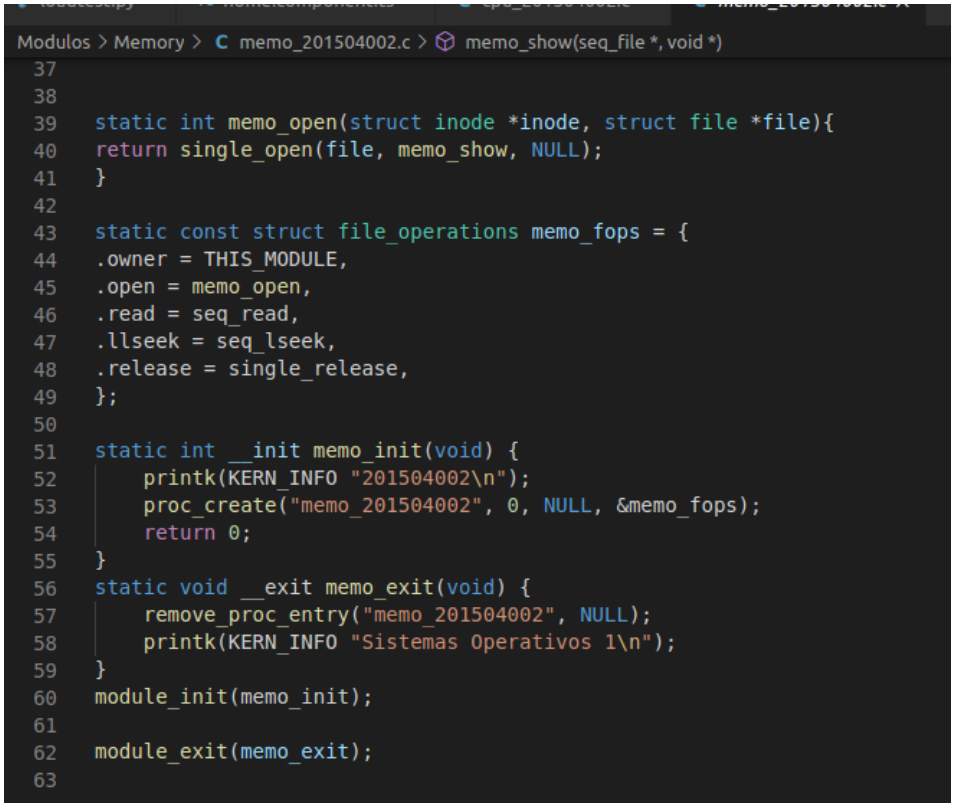
Para el módulo de memoría se utilizó sysinfo para obtener la información del sistema.

Luego se imprimió en un archivo para el módulo un JSON que contiene la memoria total, la memoria libre y el porcentaje de memoria que este posee.

```
16
17  static int memo_show(struct seq_file *m, void *v){
18
19  struct sysinfo i;
20  unsigned long porcentaje;
21
22
23  seq_printf(m, "{\n");
24  seq_printf(m, "\n");
25  si_meminfo(&i);
26
27  porcentaje = i.totalram -i.freeram;
28  porcentaje = ((porcentaje*100)/i.totalram);
29
30  seq_printf(m,"\"MemTotal\":\"%8lu\",\n",(i.totalram*4)/1024);
31  seq_printf(m,"\"MemFree\":\"%8lu\", \n",(i.freeram*4)/1024);
32  seq_printf(m,"\"MemPercent\":\"%8lu\"\n",porcentaje);
33
34  seq_printf(m, "\n}");
35  return 0;
36  }
37
38
39 > static int memo_open(struct inode *inode, struct file *file){…
42
43  static const struct file_operations memo_fops = {
44  .owner = THIS_MODULE,
45  .open = memo_open,
46  .read = seq_read,
Go 1.15.6   Select Postgres Server                                Ln 34, C
```

El memo_open se utiliza para abrir el archivo y luego file_operations se sobreescribe para que escriba en el módulo de memoria que se está creando

Luego se utiliza printk para escribir en el buffer de /proc



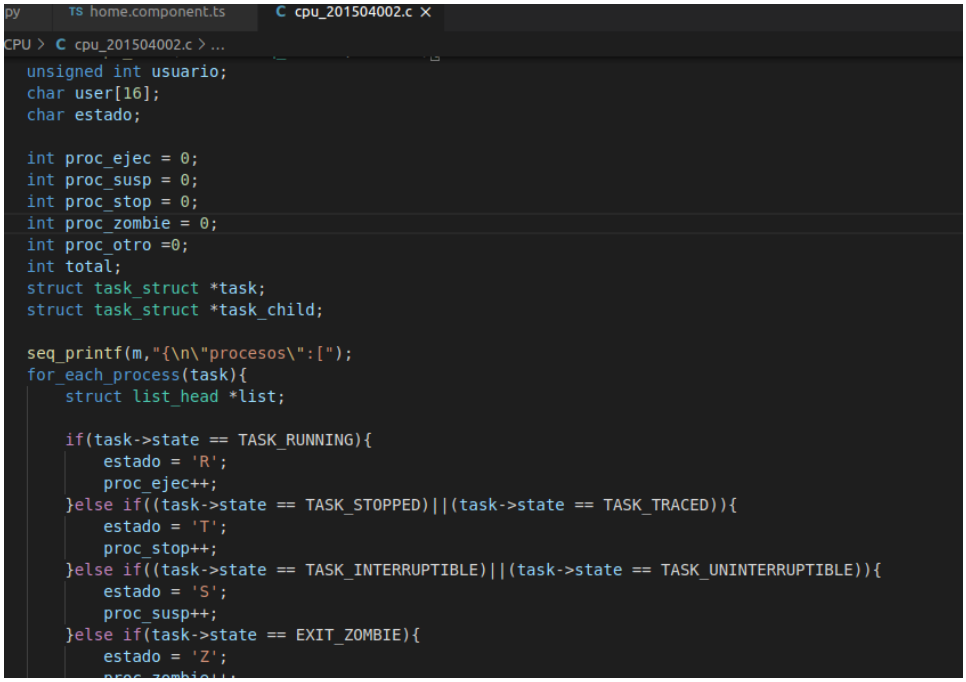
Módulo de CPU (task_struct)

Se utilizó el struct task_struct para obtener el proceso

Luego utilizando un for_each_process se recorrieron todos los tasks padres.

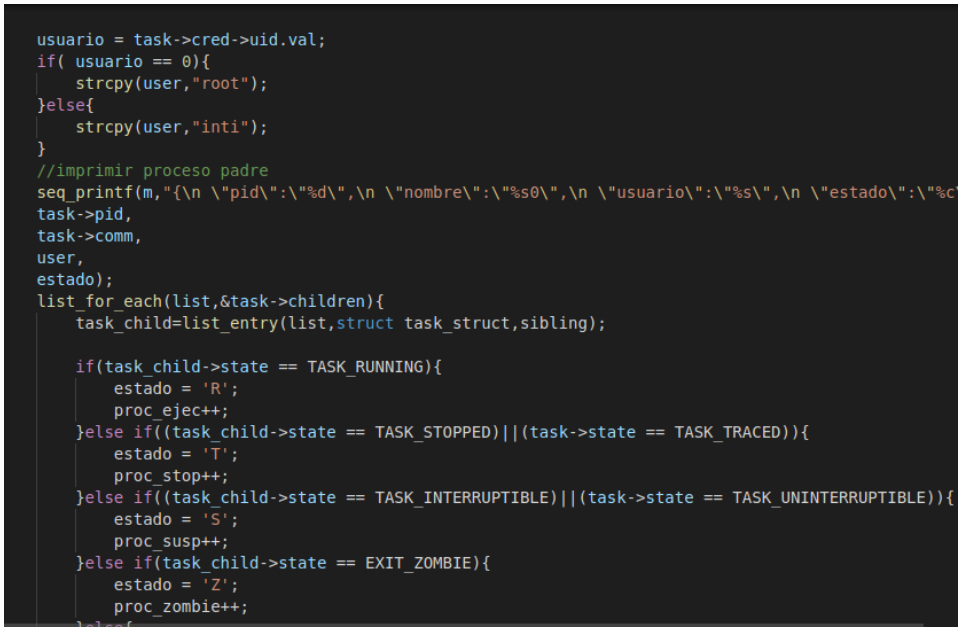
Se dividieron los estados de los procesos de la siguiente manera:

- R (Ejecutando)
- T (Detenido)
- S (Suspendido)
- Z (zombie)
- X (Otro)

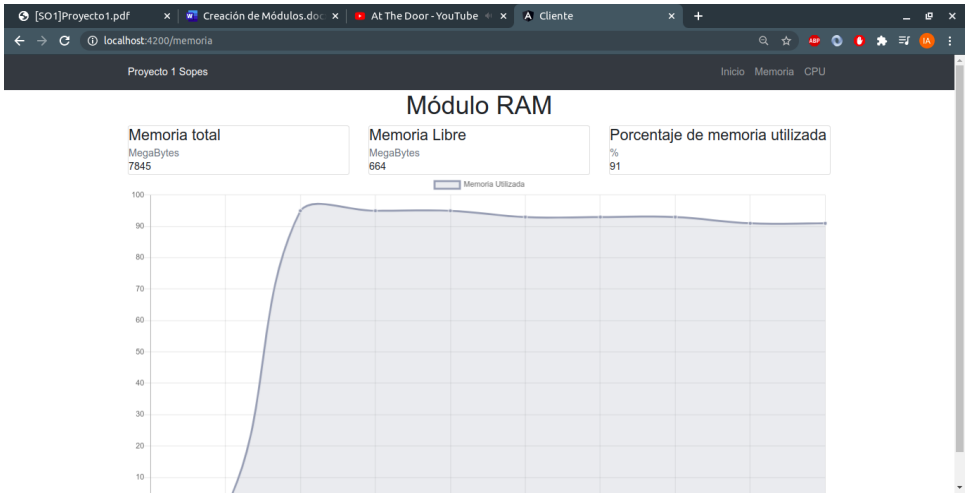


Luego se escribió el JSON del proceso.

Una vez se lee el padre, se verifica si tiene hijos con list_for_each y luego se realiza el mismo procedimiento de los estados.



Memoria



Referencias

https://www.elconspirador.com/2014/12/21/crear-un-proceso-en-un-modulo-del-kernel/

https://blog.sourcerer.io/writing-a-simple-linux-kernel-module-d9dc3762c234?gi=433ef60d45cc