

## Manual Proyecto 1

```
printk(KERN_INFO "Diciembre 2020.\n");
```

esto sirve para mostrar mensajes el log de mensajes de los módulos

```
proc_create("cpu_201403904", 0, NULL, &myops);
```

este comando es para crear un archivo en la carpeta /proc, se le envia el nombre y la referencia del struct de configuración.

```
static struct proc_ops myops =  
{  
    .proc_open = myOpen,  
    .proc_read = seq_read,  
};
```

este struct de configuración es donde se definen qué métodos se quieren sobrescribir para el módulo de manejo de archivos del kernel para el manejo del archivo específico que editaremos, únicamente se reescribió el método de abrir.

```
static int load_file (struct seq_file *m, void *v) {
```

este es el método que tiene por parámetros el archivo y n void que es el que se escribirá la información que deseamos cambie al abrir el archivo

```
seq_printf(m, "{\\"cpu\\" : {\n"});
```

esta librería se utiliza para escribir en el archivo que sea especificado, recibe como parámetro el archivo a escribir y el contenido que se desea escribir

```
for_each_process(task) {
```

se utiliza para recorrer los procesos

```
Tsize = ((get_mm_rss(task->mm)) << (PAGE_SHIFT - 10));
```

de como resultado la cantidad de memoria que esta siendo usada por el proceso que se está leyendo

```
task_child = list_entry(list, struct task_struct, sibling);
```

este comando ayuda a obtener los procesos hijos que tiene un proceso

```
#include <linux/sched.h>  
#include <linux/sched/signal.h>  
#include <linux/sched/task.h>  
#include <linux/moduleparam.h>
```

```

#include <linux/fs.h>
#include <linux/seq_file.h>
#include <linux/proc_fs.h>
#include <asm/uaccess.h>
#include <linux/init.h>
#include <linux/mm.h>

#include <linux/list.h>

#include <linux/kernel.h>
#include <linux/module.h>

```

estas son las librerías que fue necesario utilizar para el archivo de cpu

```

struct task_struct *task, *curr, *idle, *stop; //Pointer to the task
whose info will be printed
int procX = 0, procS = 0, procD = 0, procZ = 0 , procT = 0;
struct task_struct *task_child;
struct list_head *list;

```

estos fueron los struct que fueron usados para el cpu

```

#include <linux/module.h>
#include <linux/moduleparam.h>
#include <linux/init.h>
#include <linux/kernel.h>
#include <linux/proc_fs.h>
#include <linux/uaccess.h>
#include <linux/fs.h>
#include <linux/utsname.h>
#include <linux/mm.h>
#include <linux/swapfile.h>
#include <linux/seq_file.h>
#define BUFSIZE 1000

struct sysinfo i;

```

estas son las librerías que fueron usadas para el modulo de ram

```

si_meminfo(&i);
// unsigned long totalSize = ((get_mm_rss(i.to)) << (PAGE_SHIFT -
10));
seq_printf(m, "\n \"ram\" : {\"total\":%ld, \"libre\": %ld}\n",
i.totalram, i.freeram);
return 0

```

aqui esta la libreria que se uso para obtener la info de la memoria ram.