

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Sistemas Operativos 2 Sec. A
Ing. Cesar Batz
Aux Franklin



MANUAL TECNICO

Nombre:

Jeannira del Rosario Sic Menéndez
Fernando Vidal Ruiz Piox

Carnet:

201602434
201503984

Descripción

Se trata de desarrollar una aplicación que permita monitorear y gestionar los procesos de un servidor Linux, por medio de una interfaz web de fácil acceso desde el navegador de una computadora o de un dispositivo móvil como teléfono o Tablet.

La información de los procesos, uso de memoria RAM y uso de CPU será extraída mediante la implementación de módulos kernel que estarán adquiriendo y exponiendo esta información utilizando archivos ubicados en /proc.

Directorio /proc/

Grupos comunes de información referente al kernel agrupado en directorios y subdirectorios en /proc/.

> cd /proc

cada carpeta con número es un proceso. En esta práctica crearemos nuestros propios módulos.

```
ferulz@ferulz-Aspire-E5-473: /proc$ ls
1      13      1540    1982    2133    2321    2888    3019    3112    3371    38      5407    639     7060    7411    buddyinfo  kcore     schedstat
10     1313    155     2        2135    24      2893    3023    3116    3372    3854    5582    6410    7085    7475     bus        keys       scsi
1022   136    156     2003    2136    242     2895    3047    3142    3375    39      5621    6449    7122    7487     cgroups   key-users  self
1024   137    157     2017    2138    243     29      3052    3160    3395    392     5741    6462    7181    7506     cmdline   kmsg       slabinfo
1035   138    16       2022    2148    244     2913    3056    3167    34      3940    585     6504    7182    7508     consoles  kpagecgroup softirqs
1044   139    160     2024    2150    245     2920    3064    3188    3402    4        586     6518    7193    7509     cpuinfo    kpagecount stat
1047   14     1603    2028    2151    246     2925    3066    3191    3487    40      588     6536    7199    7520     crypto     kpageflags  swaps
1060   140    1604    2029    2152    27      2937    3067    3204    35      41      590     6539    7216    7532     devices    loadavg     sys
1065   141    1660    2079    2154    2754    2942    3070    3221    3500    417     594     657     7224    7533     diskstats  locks       sysrq-trigger
1088   142    1662    2081    2159    2757    2954    3073    3228    3506    42      602     660     7271    7587     dma         mdstat      sysvipc
1097   1428    1664    2086    2161    2758    2958    3077    3238    3512    427     6034    661     7285    7601     driver      meminfo     thread-self
11      143     17      2088    2164    2771    2960    3083    3254    3518    4288    6044    6629    7299    7619     execdomains misc         timer_list
1111   1439    171     2091    2171    2775    2964    3087    3285    354     429     613     6669    7307    9      fb          modules     tty
1113   144     1742    21      2195    2777    2968    3088    33      355     430     6174    671     7308    911     filesystems mounts       uptime
1115   1452    1760    2118    22      2786    2978    3095    333     36      432     6285    673     7311    912     fs          mtrr        version
1118   1479    18      2120    2236    2789    3      3100    3355    3632    4341    629     675     7378    977     interrupts net          version_signature
1121   15      1859    2121    23      28      30      3101    3362    3634    4349    632     677     7379    989     ionon       pagetypeinfo vmallocinfo
1122   150    1860    2127    2301    285     3006    3105    3363    37      4475    633     6777    7406    994     ioports     partitions  vmstat
1130   151    190     2129    2304    286     3010    3108    3366    3720    4492    635     6782    7407    acpi        irq         zoneinfo
12     152     1942    2132    2313    2885    3015    3109    3369    378     4533    637     679     7408    asound      kallsyms    sched_debug
ferulz@ferulz-Aspire-E5-473: /proc$
```

Primera Parte del Proyecto

Módulo de memoria (sysinfo)

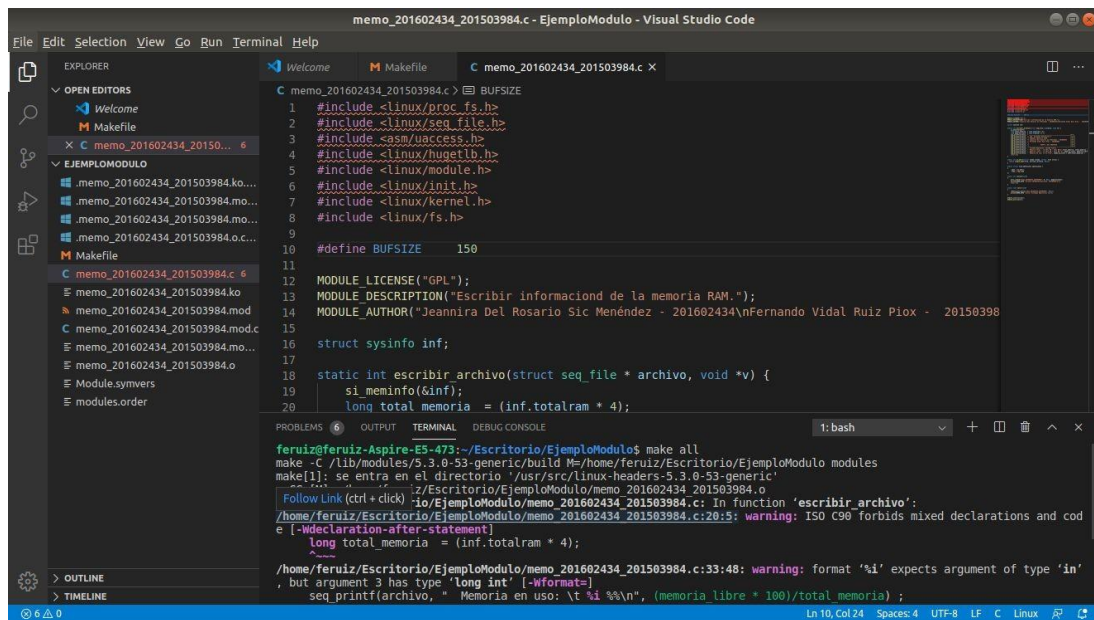
crear dos archivos

Makefile mem_grupo33.c

Luego se utilizan los siguiente comandos:

Para crear los archivos necesarios. (el que me importa es el archivo .ko)

> make all



```
File Edit Selection View Go Run Terminal Help
memo_201602434_201503984.c - EjemploModulo - Visual Studio Code

EXPLORER
  OPEN EDITORS
    Welcome
  Makefile
  memo_201602434_201503984.c
  EJEEMPLOMODULO
    .memo_201602434_201503984.ko
    .memo_201602434_201503984.mo
    .memo_201602434_201503984.mo
    .memo_201602434_201503984.o.c
    Makefile
    C memo_201602434_201503984.c
    memo_201602434_201503984.ko
    memo_201602434_201503984.mod
    C memo_201602434_201503984.mod.c
    E memo_201602434_201503984.mo
    E memo_201602434_201503984.o
    Module.symvers
    modules.order

C memo_201602434_201503984.c
1 #include <linux/proc_fs.h>
2 #include <linux/seq_file.h>
3 #include <asm/uaccess.h>
4 #include <linux/hugetlb.h>
5 #include <linux/module.h>
6 #include <linux/init.h>
7 #include <linux/kernel.h>
8 #include <linux/fs.h>
9
10 #define BUFSIZE 150
11
12 MODULE_LICENSE("GPL");
13 MODULE_DESCRIPTION("Escribir informacion de la memoria RAM.");
14 MODULE_AUTHOR("Jeannira Del Rosario Sic Menéndez - 201602434\Fernando Vidal Ruiz Piox - 20150398");
15
16 struct sysinfo inf;
17
18 static int escribir_archivo(struct seq_file * archivo, void *v) {
19     si_meminfo(&inf);
20     long total_memoria = (inf.totalram * 4);
21
22     /home/feruiz/Escritorio/EjemploModulo/memo_201602434_201503984.c:33:48: warning: ISO C90 forbids mixed declarations and code [-Wdeclaration-after-statement]
23         long total_memoria = (inf.totalram * 4);
24
25     /home/feruiz/Escritorio/EjemploModulo/memo_201602434_201503984.c:33:48: warning: format '%i' expects argument of type 'int', but argument 3 has type 'long int' [-Wformat=]
26     seq_printf(archivo, " Memoria en uso: \t%i %\n", (memoria libre * 100)/total_memoria) ;
27 }
28
29 int __init escribir_archivo_init(void) {
30     return 0;
31 }
32
33 MODULE_ALIAS("mem_grupo33");
34
35
```

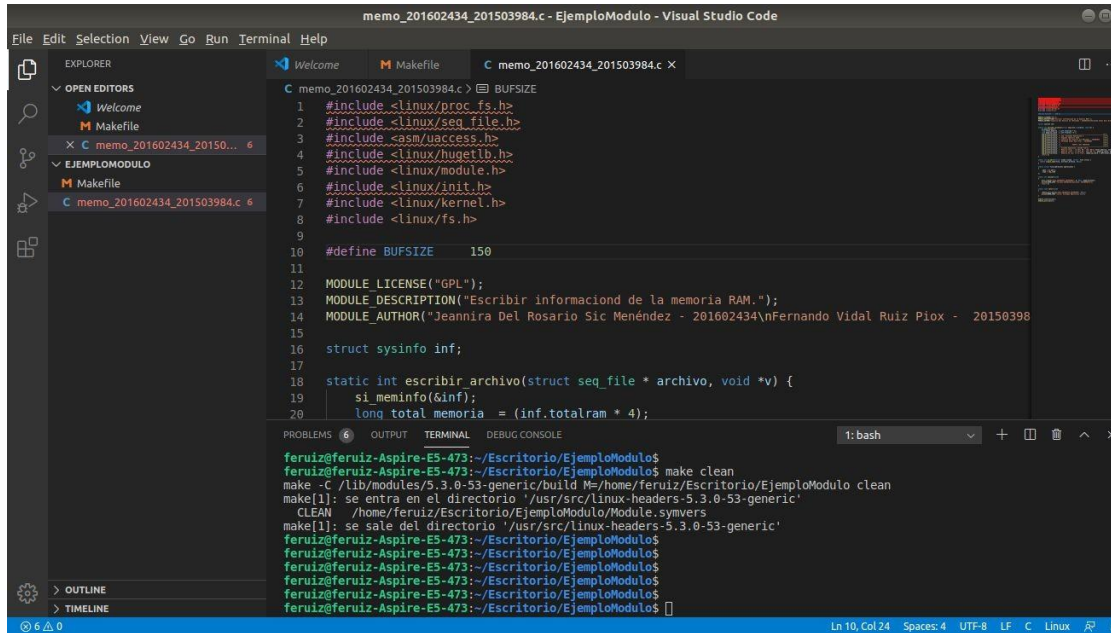
```
feruiz@feruiz-Aspire-E5-473:~/Escritorio/EjemploModulo$ make all
make -C /lib/modules/5.3.0-53-generic/build M=/home/feruiz/Escritorio/EjemploModulo modules
make[1]: se entra en el directorio /usr/src/linux-headers-5.3.0-53-generic
...
/home/feruiz/Escritorio/EjemploModulo/memo_201602434_201503984.o
/home/feruiz/Escritorio/EjemploModulo/memo_201602434_201503984.c:20:5: warning: ISO C90 forbids mixed declarations and code [-Wdeclaration-after-statement]
     long total_memoria = (inf.totalram * 4);
     ^~~~~
/home/feruiz/Escritorio/EjemploModulo/memo_201602434_201503984.c:33:48: warning: format '%i' expects argument of type 'int', but argument 3 has type 'long int' [-Wformat=]
     seq_printf(archivo, " Memoria en uso: \t%i %\n", (memoria libre * 100)/total_memoria) ;
                                                ^~

```

Para limpiar los archivos que se crearon al ejecutar el comando anterior

> make clean

Este comando sirve para insertar el módulo



The screenshot shows the Visual Studio Code editor with the file `memo_201602434_201503984.c` open. The code is a C program for a kernel module. It includes headers like `<linux/proc_fs.h>`, `<linux/seq_file.h>`, `<asm/uaccess.h>`, `<linux/hugetlb.h>`, `<linux/module.h>`, `<linux/init.h>`, `<linux/kernel.h>`, and `<linux/fs.h>`. It defines `BUFSIZE` as 150 and has a `MODULE_LICENSE("GPL")`. The `MODULE_DESCRIPTION` is "Escribir informacion de la memoria RAM." and the `MODULE_AUTHOR` is "Jeannira Del Rosario Sic Menéndez - 201602434\nFernando Vidal Ruiz Piox - 20150398". The code defines a `sysinfo` struct and a function `escribir_archivo` that writes to a file. The terminal shows the execution of `make clean`, which removes the module files.

```
feruiz@feruiz-Aspire-E5-473:~/Escritorio/EjemploModulo$
feruiz@feruiz-Aspire-E5-473:~/Escritorio/EjemploModulo$ make clean
make -C /lib/modules/5.3.0-53-generic/build M=/home/feruiz/Escritorio/EjemploModulo clean
make[1]: se entra en el directorio '/usr/src/linux-headers-5.3.0-53-generic'
CLEAN /home/feruiz/Escritorio/EjemploModulo/Module.symvers
make[1]: se sale del directorio '/usr/src/linux-headers-5.3.0-53-generic'
feruiz@feruiz-Aspire-E5-473:~/Escritorio/EjemploModulo$
feruiz@feruiz-Aspire-E5-473:~/Escritorio/EjemploModulo$
feruiz@feruiz-Aspire-E5-473:~/Escritorio/EjemploModulo$
feruiz@feruiz-Aspire-E5-473:~/Escritorio/EjemploModulo$
feruiz@feruiz-Aspire-E5-473:~/Escritorio/EjemploModulo$
feruiz@feruiz-Aspire-E5-473:~/Escritorio/EjemploModulo$
feruiz@feruiz-Aspire-E5-473:~/Escritorio/EjemploModulo$
```

> sudo insmod `mem_grupo33.ko`

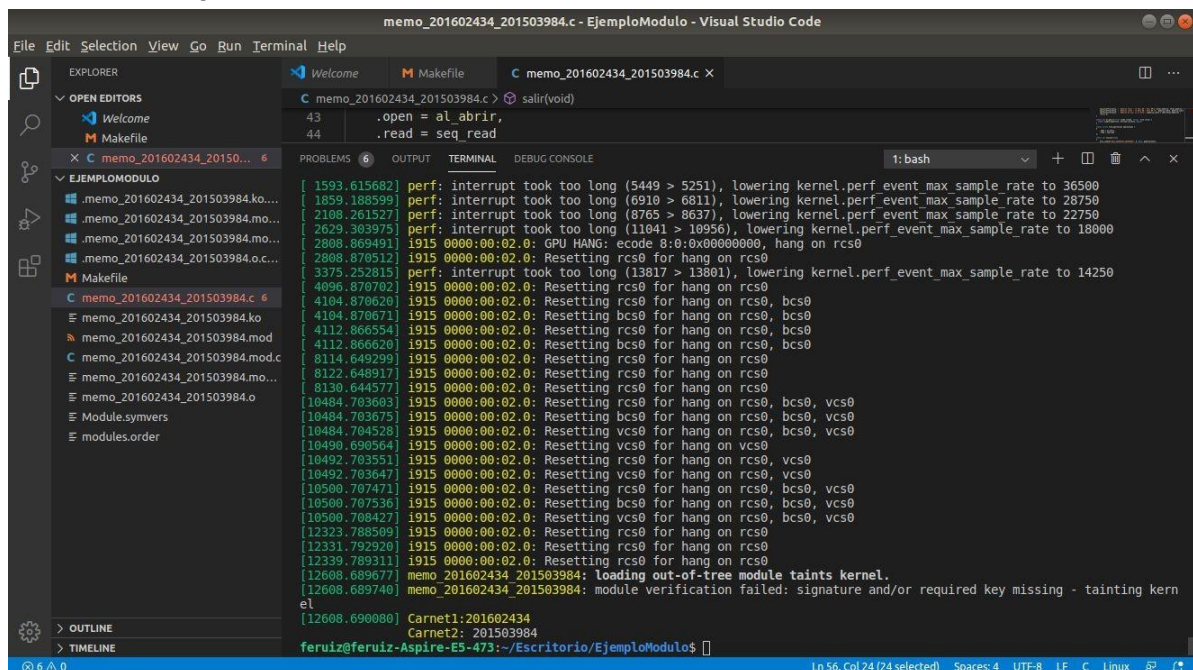
Este comando sirve para remover un módulo

> sudo rmmod `mem_grupo33.ko`

Muestra todos los mensajes que están mostrando los módulos.

En este caso, de último se mostrarán los números de carnet (es el mensaje que se muestra al iniciar nuestro módulo).

> dmesg



The screenshot shows the Visual Studio Code editor with the file `memo_201602434_201503984.c` open. The terminal shows the output of the `dmesg` command, which displays kernel messages. The messages include performance warnings, GPU hang warnings, and module loading messages. The last message shows the module verification failed due to a missing signature and/or required key.

```
[ 1593.615682] perf: interrupt took too long (5449 > 5251), lowering kernel.perf_event_max_sample_rate to 36500
[ 1859.188599] perf: interrupt took too long (6910 > 6811), lowering kernel.perf_event_max_sample_rate to 28750
[ 2108.261527] perf: interrupt took too long (8765 > 8637), lowering kernel.perf_event_max_sample_rate to 22750
[ 2629.303975] perf: interrupt took too long (11041 > 10956), lowering kernel.perf_event_max_sample_rate to 18000
[ 2808.869491] 1915 0000:00:02.0: GPU HANG: ecode 8:0:0x00000000, hang on rcs0
[ 2808.870512] 1915 0000:00:02.0: Resetting rcs0 for hang on rcs0
[ 3375.252015] perf: interrupt took too long (13817 > 13801), lowering kernel.perf_event_max_sample_rate to 14250
[ 4096.870702] 1915 0000:00:02.0: Resetting rcs0 for hang on rcs0
[ 4104.870620] 1915 0000:00:02.0: Resetting rcs0 for hang on rcs0, bcs0
[ 4104.870671] 1915 0000:00:02.0: Resetting bcs0 for hang on rcs0, bcs0
[ 4112.866554] 1915 0000:00:02.0: Resetting rcs0 for hang on rcs0, bcs0
[ 4112.866620] 1915 0000:00:02.0: Resetting bcs0 for hang on rcs0, bcs0
[ 8114.649299] 1915 0000:00:02.0: Resetting rcs0 for hang on rcs0
[ 8122.648917] 1915 0000:00:02.0: Resetting rcs0 for hang on rcs0
[ 8130.644577] 1915 0000:00:02.0: Resetting rcs0 for hang on rcs0
[10484.703603] 1915 0000:00:02.0: Resetting rcs0 for hang on rcs0, bcs0, vcs0
[10484.703675] 1915 0000:00:02.0: Resetting bcs0 for hang on rcs0, bcs0, vcs0
[10484.704528] 1915 0000:00:02.0: Resetting vcs0 for hang on rcs0, bcs0, vcs0
[10490.690564] 1915 0000:00:02.0: Resetting vcs0 for hang on rcs0, vcs0
[10492.703551] 1915 0000:00:02.0: Resetting rcs0 for hang on rcs0, vcs0
[10492.703647] 1915 0000:00:02.0: Resetting vcs0 for hang on rcs0, vcs0
[10500.707471] 1915 0000:00:02.0: Resetting rcs0 for hang on rcs0, bcs0, vcs0
[10500.707536] 1915 0000:00:02.0: Resetting bcs0 for hang on rcs0, bcs0, vcs0
[10500.708427] 1915 0000:00:02.0: Resetting vcs0 for hang on rcs0, bcs0, vcs0
[12323.788509] 1915 0000:00:02.0: Resetting rcs0 for hang on rcs0
[12331.792920] 1915 0000:00:02.0: Resetting rcs0 for hang on rcs0
[12339.780311] 1915 0000:00:02.0: Resetting rcs0 for hang on rcs0
[12608.689677] memo_201602434_201503984: loading out-of-tree module taints kernel.
[12608.689740] memo_201602434_201503984: module verification failed: signature and/or required key missing - tainting kern
[12608.690000] Carnet1:201602434
[12608.690000] Carnet2:201503984
feruiz@feruiz-Aspire-E5-473:~/Escritorio/EjemploModulo$
```

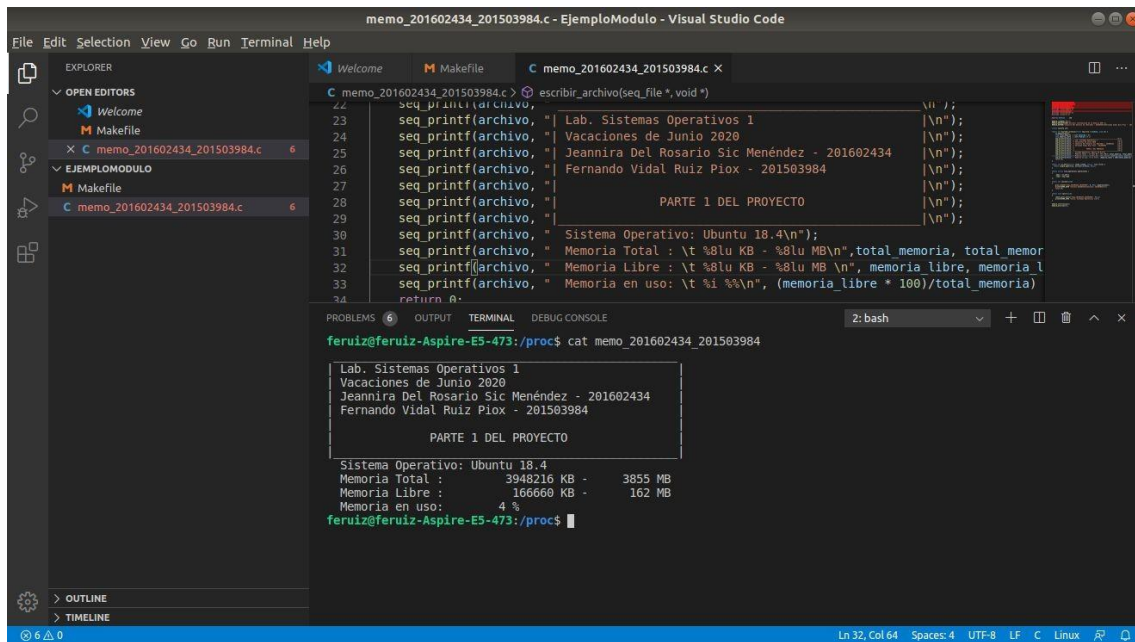
Luego abrimos una nueva terminal e ingresamos a la carpeta proc

```
> cd /proc
```

Ingresamos el siguiente comando para ver el contenido del archivo

“mem_grupo33”, cabe resaltar que cada vez que ejecutamos este comando, la información al leer el archivo se actualiza.

```
> cat mem_grupo33
```



The screenshot shows the Visual Studio Code editor with a C program named `memo_201602434_201503984.c` open. The program uses `seq_printf` to write data to a file. The terminal shows the output of the `cat` command, displaying the contents of the file. The output includes a header, a list of names, a section titled "PARTE 1 DEL PROYECTO", and system memory statistics.

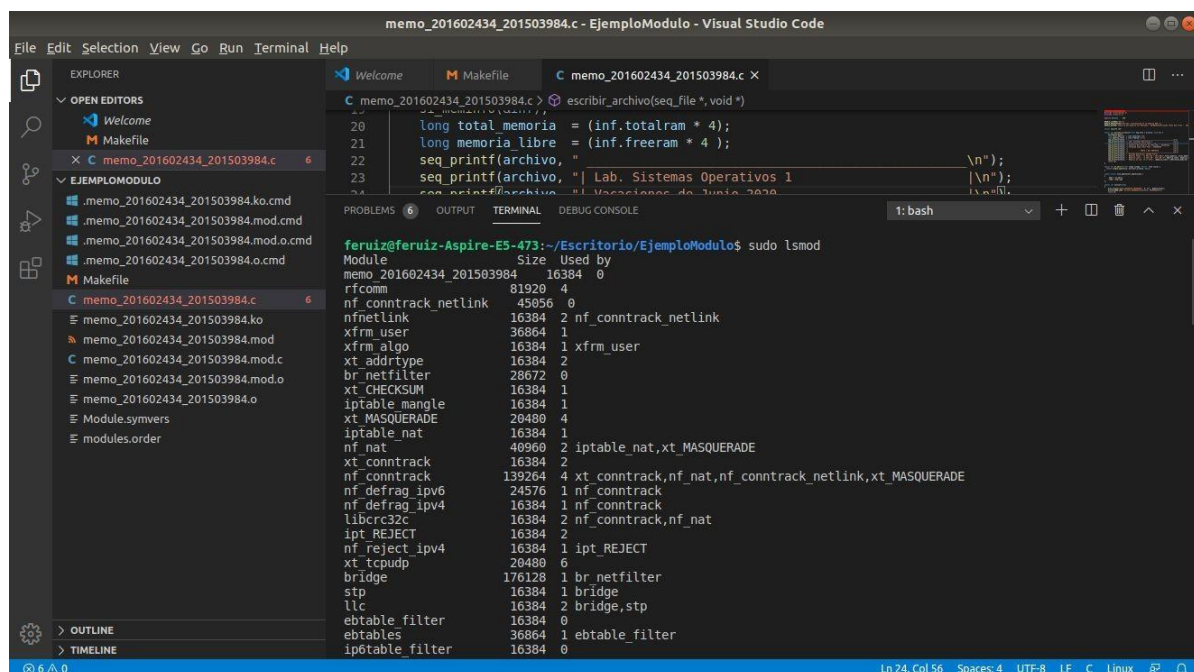
```
Lab. Sistemas Operativos 1
Vacaciones de Junio 2020
Jeannira Del Rosario Sic Menéndez - 201602434
Fernando Vidal Ruiz Piox - 201503984

PARTE 1 DEL PROYECTO

Sistema Operativo: Ubuntu 18.4
Memoria Total : 3948216 KB - 3855 MB
Memoria Libre : 166660 KB - 162 MB
Memoria en uso: 4 %
```

Este comando sirve para listar todos los modulo

```
> sudo lsmod
```



The screenshot shows the Visual Studio Code editor with the same C program open. The terminal displays the output of the `lsmod` command, listing loaded kernel modules and their dependencies.

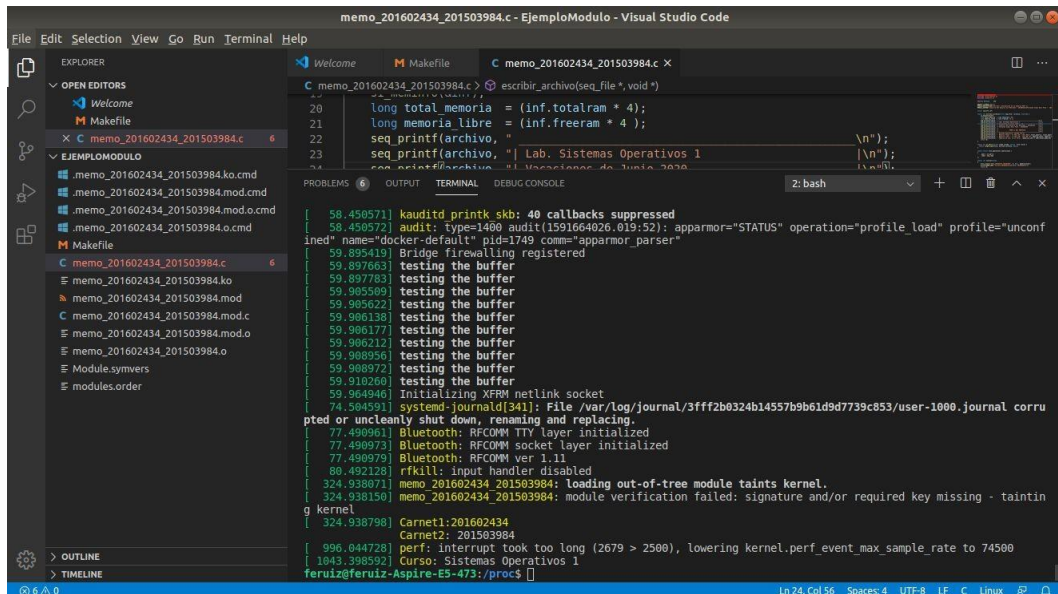
```
Module      Size  Used by
memo_201602434_201503984 16384 0
rfcomm      81920 4
nf_conntrack_netlink 45856 0
nfnetlink   16384 2 nf_conntrack_netlink
xfrm_user   36864 1
xfrm_algo   16384 1 xfrm_user
xt_addrtype 16384 2
br_netfilter 28672 0
xt_CHECKSUM 16384 1
iptable_mangle 16384 1
xt_MASQUERADE 20480 4
iptables_nat 16384 1
nf_nat      40960 2 iptable_nat,xt_MASQUERADE
xt_conntrack 16384 2
nf_conntrack 139264 4 xt_conntrack,nf_nat,nf_conntrack_netlink,xt_MASQUERADE
nf_defrag_ipv6 24576 1 nf_conntrack
nf_defrag_ipv4 16384 1 nf_conntrack
libcrc32c    16384 2 nf_conntrack,nf_nat
ipt_REJECT   16384 2
nf_reject_ipv4 16384 1 ipt_REJECT
xt_tcpudp    20480 6
bridge      176128 1 br_netfilter
stp          16384 1 bridge
llc          16384 2 bridge,stp
ebtable_filter 36864 1 ebtable_filter
ebtables     36864 1 ebtable_filter
ip6table_filter 16384 0
```


removemos el modulo

```
> sudo rmmod mem_grupo33.ko
```

vemos el mensaje que muestra al ser removido el módulo.

```
> dmesg
```



The screenshot shows the Visual Studio Code interface with the 'memo_201602434_201503984.c' file open. The terminal window displays the output of the 'dmesg' command, showing the removal of the 'memo' module. The log includes messages from the kernel, such as 'kuditt printk skb: 40 callbacks suppressed', 'audit: type=1400 audit(1591664026.019:52): apparmor="STATUS" operation="profile_load" profile="unconfined" name="docker-default" pid=1749 comm="apparmor_parser"', and 'Bridge firewalling registered'. It also shows the module's internal state, including 'testing the buffer' and 'Initializing XFRM netlink socket'. The final message is 'memo_201602434_201503984: Loading out-of-tree module taints kernel.' followed by 'kernel' and 'Carnet1:201602434'.

Módulo de CPU (task_struct)

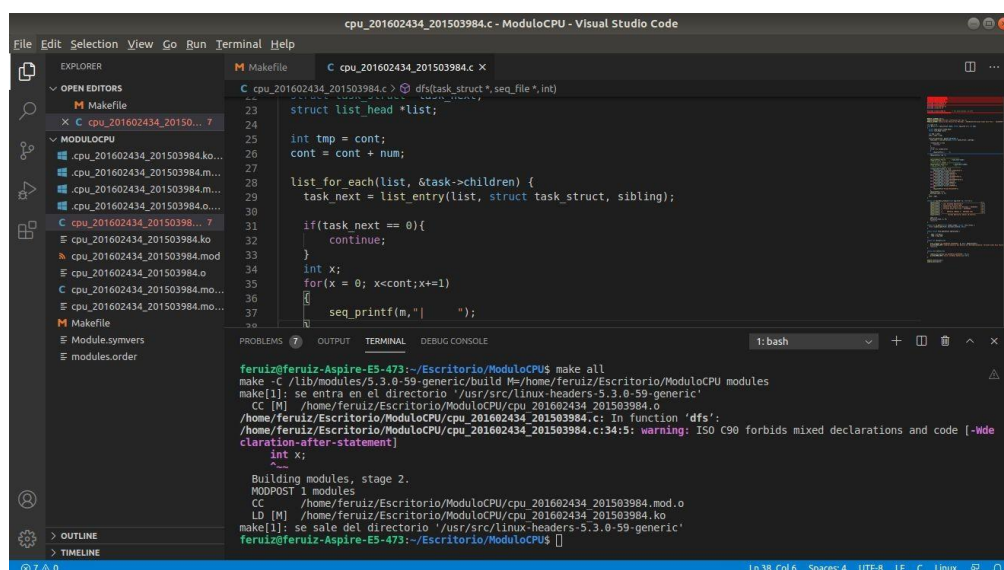
crear dos archivos

Makefile cpu_grupo33.c

Luego se utilizan los siguiente comandos:

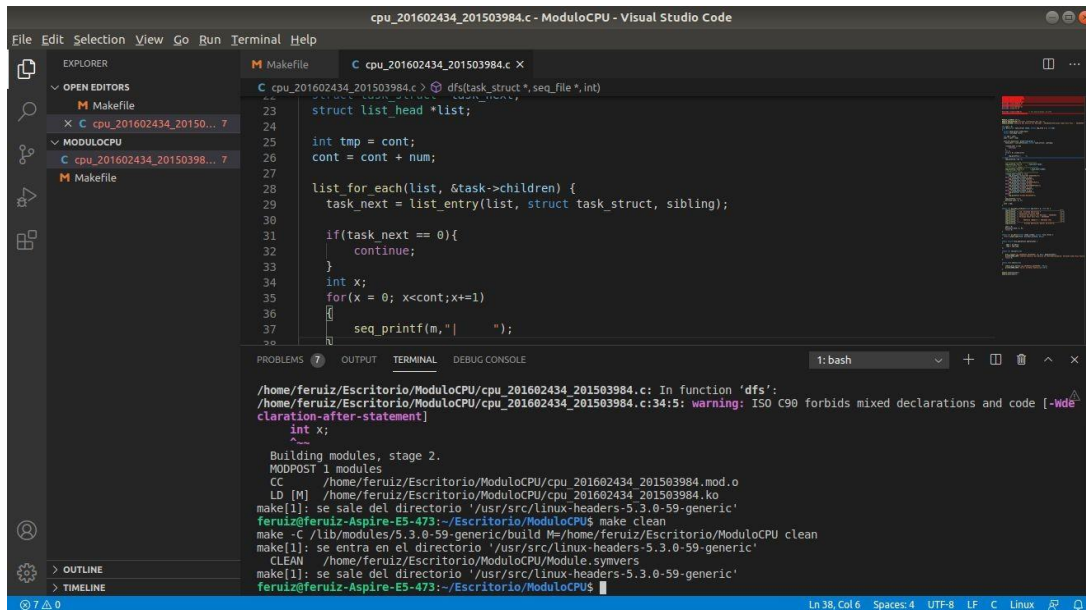
Para crear los archivos necesarios. (agregará 11 archivos más pero el que nos importará es el archivo .ko)

```
> make all
```



The screenshot shows the Visual Studio Code interface with the 'cpu_201602434_201503984.c' file open. The terminal window displays the output of the 'make all' command, showing the removal of the 'cpu' module. The log includes messages from the kernel, such as 'Building modules, stage 2.', 'MODPOST 1 modules', and 'CC [M] /home/feruiz/Escritorio/ModuloCPU/cpu_201602434_201503984.o'. It also shows the module's internal state, including 'Building modules, stage 2.', 'MODPOST 1 modules', and 'CC [M] /home/feruiz/Escritorio/ModuloCPU/cpu_201602434_201503984.o'. The final message is 'cpu_201602434_201503984: Loading out-of-tree module taints kernel.' followed by 'kernel' and 'Carnet1:201602434'.

Para limpiar los archivos que se crearon al ejecutar el comando anterior
> make clean



The screenshot shows the Visual Studio Code interface with a C program in the editor and its compilation output in the terminal. The C program defines a linked list structure and a function to traverse it. The terminal shows the compilation process, including the creation of object files and the final executable. A warning is displayed: 'warning: ISO C90 forbids mixed declarations and code [-Wdeclaration-after-statement]'. The compilation is successful, and the terminal shows the command 'make clean' being executed.

```
File Edit Selection View Go Run Terminal Help
cpu_201602434_201503984.c - ModuloCPU - Visual Studio Code

EXPLORER
  OPEN EDITORS
    Makefile
    cpu_201602434_201503984.c
    ModuloCPU
    cpu_201602434_201503984.c
    Makefile

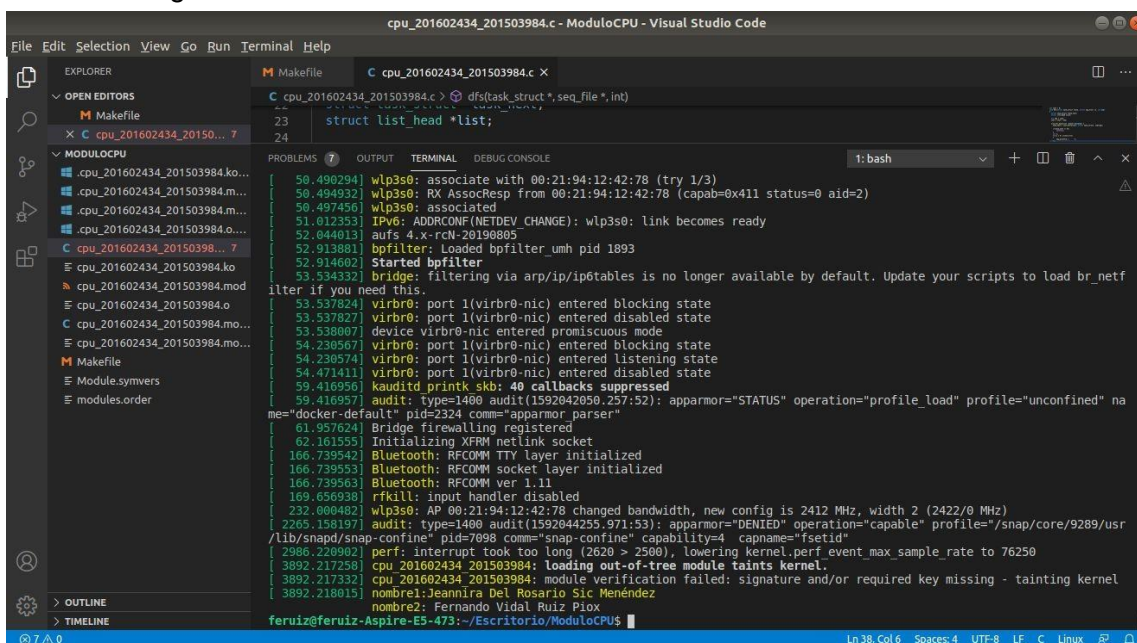
  PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
    1: bash

/home/feruiz/Escritorio/ModuloCPU/cpu_201602434_201503984.c: In function 'dfs':
/home/feruiz/Escritorio/ModuloCPU/cpu_201602434_201503984.c:34:5: warning: ISO C90 forbids mixed declarations and code [-Wdeclaration-after-statement]
    int x;
    ^
Building modules, stage 2.
MODPOST 1 modules
CC /home/feruiz/Escritorio/ModuloCPU/cpu_201602434_201503984.mod.o
LD [M] /home/feruiz/Escritorio/ModuloCPU/cpu_201602434_201503984.ko
make[1]: se sale del directorio '/usr/src/linux-headers-5.3.0-59-generic'
feruiz@feruiz-Aspire-E5-473:~/Escritorio/ModuloCPU$ make clean
make -C /lib/modules/5.3.0-59-generic/build M=/home/feruiz/Escritorio/ModuloCPU clean
make[1]: se entra en el directorio '/usr/src/linux-headers-5.3.0-59-generic'
CLEAN /home/feruiz/Escritorio/ModuloCPU/Module.symvers
make[1]: se sale del directorio '/usr/src/linux-headers-5.3.0-59-generic'
feruiz@feruiz-Aspire-E5-473:~/Escritorio/ModuloCPU$
```

Este comando sirve para insertar el módulo
> sudo insmod **cpu_grupo33.ko**

Este comando sirve para remover un módulo
> sudo rmmod **cpu_grupo33.ko**

Después ejecutaremos el siguiente comando, para mostrar el mensaje del modulo.
En este caso, de último se mostrarán los nombres de los integrantes del grupo (es el mensaje que se muestra al iniciar nuestro módulo).
> dmesg



The screenshot shows the Visual Studio Code interface with the terminal displaying the output of the 'dmesg' command. The output shows the kernel log, including the loading of the 'cpu_grupo33.ko' module and the execution of the 'dfs' function. The terminal output is as follows:

```
File Edit Selection View Go Run Terminal Help
cpu_201602434_201503984.c - ModuloCPU - Visual Studio Code

EXPLORER
  OPEN EDITORS
    Makefile
    cpu_201602434_201503984.c
    ModuloCPU
    cpu_201602434_201503984.c
    Makefile

  PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
    1: bash

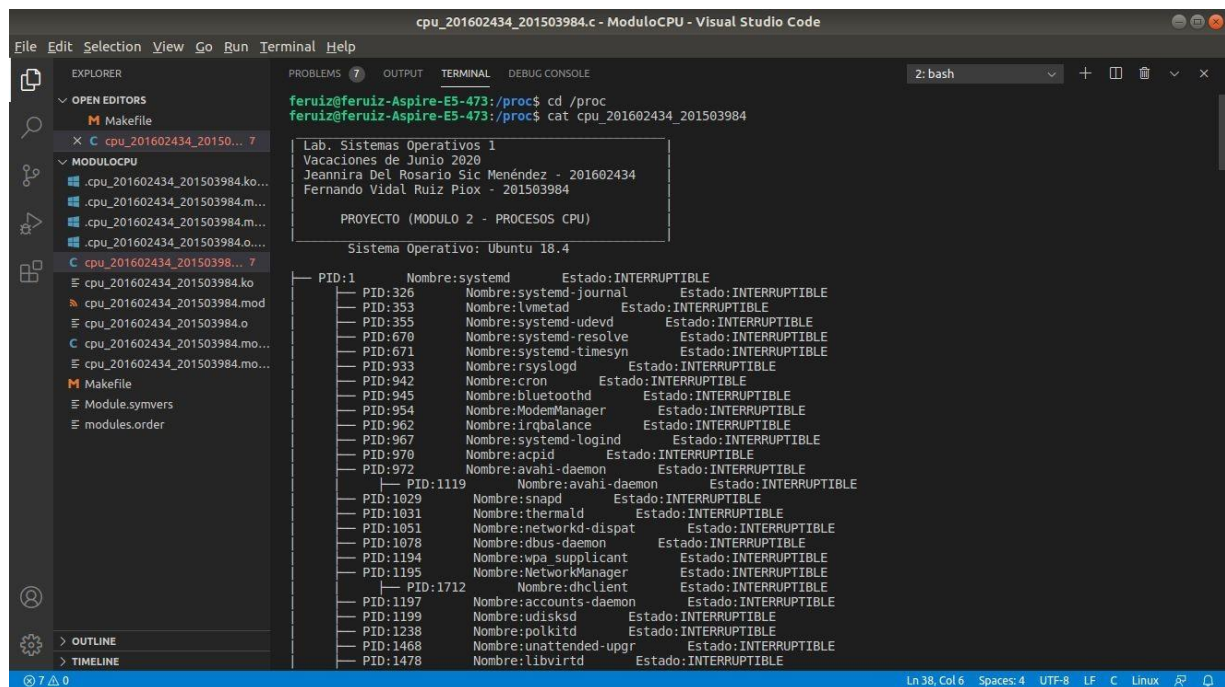
[ 50.490294] wlp3s0: associate with 00:21:94:12:42:78 (try 1/3)
[ 50.497456] wlp3s0: RX AssocResp from 00:21:94:12:42:78 (capab=0x411 status=0 aid=2)
[ 50.497456] wlp3s0: associated
[ 51.012353] IPv6: ADDRCONF(NETDEV_CHANGE): wlp3s0: link becomes ready
[ 52.044013] aufs: 4.x-rcN-20190805
[ 52.913881] bpfILTER: Loaded bpfILTER umh pid 1893
[ 52.914602] Started bpfILTER
[ 53.534332] bridge: filtering via arp/ip6tables is no longer available by default. Update your scripts to load br_netf
[ 53.537824] virbr0: port 1(virbr0-nic) entered blocking state
[ 53.537827] virbr0: port 1(virbr0-nic) entered disabled state
[ 53.538007] device virbr0-nic entered promiscuous mode
[ 54.230567] virbr0: port 1(virbr0-nic) entered blocking state
[ 54.230574] virbr0: port 1(virbr0-nic) entered listening state
[ 54.471411] virbr0: port 1(virbr0-nic) entered disabled state
[ 59.416956] kauditd: print skbs: 40 callbacks suppressed
[ 59.416957] audit: type=1400 audit(1592042050.257:52): apparmor="STATUS" operation="profile_load" profile="unconfined" na
me="docker-default" pid=2324 comm="apparmor parser"
[ 61.957624] Bridge firewalling registered
[ 62.161555] Initializing XFRM netlink socket
[ 166.739542] Bluetooth: RFCOMM TTY layer initialized
[ 166.739553] Bluetooth: RFCOMM socket layer initialized
[ 166.739563] Bluetooth: RFCOMM ver 1.11
[ 169.656938] rtkill: input handler disabled
[ 232.000482] wlp3s0: AP 00:21:94:12:42:78 changed bandwidth, new config is 2412 MHz, width 2 (2422/0 MHz)
[ 2265.158197] audit: type=1400 audit(1592044255.971:53): apparmor="DENIED" operation="capable" profile="/snap/core/9289/usr
/lib/snapd/snap-confine" pid=7098 comm="snap-confine" capability=4 capname="fsetid"
[ 2986.220902] perf: interrupt took too long (2620 > 2500), lowering kernel.perf_event_max_sample_rate to 76250
[ 3892.217258] cpu_201602434_201503984: loading out-of-tree module taints kernel.
[ 3892.217332] cpu_201602434_201503984: module verification failed: signature and/or required key missing - tainting kernel
[ 3892.218015] nombre1: Jeannira Del Rosario Sic Menendez
[ 3892.218015] nombre2: Fernando Vidal Ruiz Piox
feruiz@feruiz-Aspire-E5-473:~/Escritorio/ModuloCPU$
```

luego abrimos una nueva terminal e ingresamos a la carpeta proc

```
> cd /proc
```

Ingresamos el siguiente comando para ver el contenido del archivo “cpu_grupo33”, cabe resaltar que cada vez que ejecutamos este comando, la información al leer el archivo se actualiza.

```
> cat cpu_grupo33
```



```
cpu_201602434_201503984.c - ModuloCPU - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
  OPEN EDITORS
    Makefile
  X cpu_201602434_20150... 7
  MODULOCPU
    .cpu_201602434_201503984.ko...
    .cpu_201602434_201503984.m...
    .cpu_201602434_201503984.m...
    .cpu_201602434_201503984.o...
  C cpu_201602434_20150398... 7
    .cpu_201602434_201503984.ko...
    .cpu_201602434_201503984.mod
    .cpu_201602434_201503984.o
    .cpu_201602434_201503984.mo...
    .cpu_201602434_201503984.mo...
  Makefile
  Module.symvers
  modules.order
  OUTLINE
  TIMELINE
PROBLEMS 7 OUTPUT TERMINAL DEBUG CONSOLE
2: bash
feruiz@feruiz-Aspire-E5-473:/proc$ cd /proc
feruiz@feruiz-Aspire-E5-473:/proc$ cat cpu_201602434_201503984

Lab. Sistemas Operativos 1
Vacaciones de Junio 2020
Jeannira Del Rosario Sic Menéndez - 201602434
Fernando Vidal Ruiz Piox - 201503984

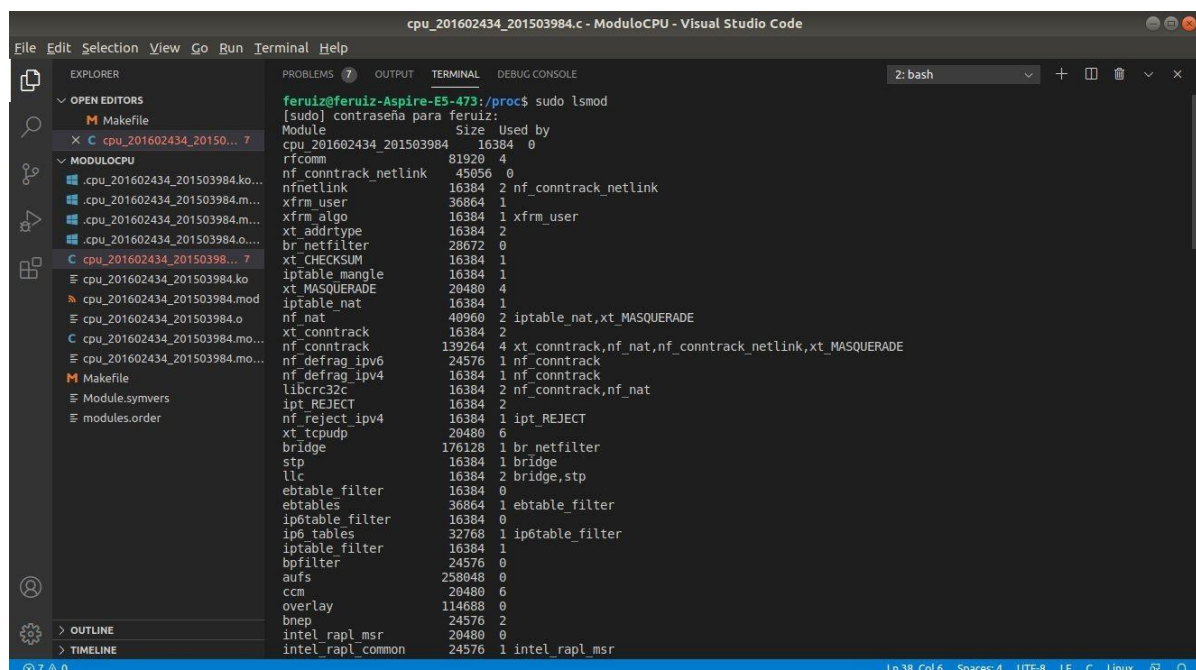
PROYECTO (MODULO 2 - PROCESOS CPU)

Sistema Operativo: Ubuntu 18.4

PID:1      Nombre:systemd      Estado:INTERRUPTIBLE
PID:326    Nombre:systemd-journal Estado:INTERRUPTIBLE
PID:353    Nombre:lvmetad       Estado:INTERRUPTIBLE
PID:355    Nombre:systemd-udev  Estado:INTERRUPTIBLE
PID:670    Nombre:systemd-resolve Estado:INTERRUPTIBLE
PID:671    Nombre:systemd-timesyn Estado:INTERRUPTIBLE
PID:933    Nombre:rsyslogd      Estado:INTERRUPTIBLE
PID:942    Nombre:cron          Estado:INTERRUPTIBLE
PID:945    Nombre:bluetoothd   Estado:INTERRUPTIBLE
PID:954    Nombre:ModemManager  Estado:INTERRUPTIBLE
PID:962    Nombre:irqbalance   Estado:INTERRUPTIBLE
PID:967    Nombre:systemd-logind Estado:INTERRUPTIBLE
PID:979    Nombre:acpid         Estado:INTERRUPTIBLE
PID:972    Nombre:avahi-daemon  Estado:INTERRUPTIBLE
  PID:1119  Nombre:avahi-daemon  Estado:INTERRUPTIBLE
PID:1029   Nombre:snappy        Estado:INTERRUPTIBLE
PID:1031   Nombre:thermald      Estado:INTERRUPTIBLE
PID:1051   Nombre:networkd-dispat Estado:INTERRUPTIBLE
PID:1078   Nombre:dbus-daemon   Estado:INTERRUPTIBLE
PID:1194   Nombre:wpa-suplicant Estado:INTERRUPTIBLE
PID:1195   Nombre:NetworkManager Estado:INTERRUPTIBLE
  PID:1712  Nombre:dhclient      Estado:INTERRUPTIBLE
PID:1197   Nombre:accounts-daemon Estado:INTERRUPTIBLE
PID:1199   Nombre:udisksd       Estado:INTERRUPTIBLE
PID:1238   Nombre:polkitd       Estado:INTERRUPTIBLE
PID:1468   Nombre:unattended-upgr Estado:INTERRUPTIBLE
PID:1470   Nombre:libvirtd      Estado:INTERRUPTIBLE
```

Este comando sirve para listar todos los modulo.

```
sudo lsmod
```



```
cpu_201602434_201503984.c - ModuloCPU - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
  OPEN EDITORS
    Makefile
  X cpu_201602434_20150... 7
  MODULOCPU
    .cpu_201602434_201503984.ko...
    .cpu_201602434_201503984.m...
    .cpu_201602434_201503984.m...
    .cpu_201602434_201503984.o...
  C cpu_201602434_20150398... 7
    .cpu_201602434_201503984.ko...
    .cpu_201602434_201503984.mod
    .cpu_201602434_201503984.o
    .cpu_201602434_201503984.mo...
    .cpu_201602434_201503984.mo...
  Makefile
  Module.symvers
  modules.order
  OUTLINE
  TIMELINE
PROBLEMS 7 OUTPUT TERMINAL DEBUG CONSOLE
2: bash
feruiz@feruiz-Aspire-E5-473:/proc$ sudo lsmod
[sudo] contraseña para feruiz:
Module      Size  Used by
cpu_201602434_201503984 16384 0
rfcomm      81920 4
nf_conntrack_netlink 45056 0
nfnetlink   16384 2 nf_conntrack_netlink
xfrm_user   36864 1
xfrm_algo   16384 1 xfrm_user
xt_addrtype 16384 2
br_netfilter 28672 0
xt_CHECKSUM 16384 1
ipTable mangle 16384 1
xt_MASQUERADE 20480 4
ipTable nat 16384 1
nf_nat      40960 2 ipTable_nat,xt_MASQUERADE
xt_conntrack 16384 2
nf_conntrack 139264 4 xt_conntrack,nf_nat,nf_conntrack_netlink,xt_MASQUERADE
nf_defrag_ipv6 24576 1 nf_conntrack
nf_defrag_ipv4 16384 1 nf_conntrack
libcrc32c   16384 2 nf_conntrack,nf_nat
ipt_REJECT 16384 2
nf_reject_ipv4 16384 1 ipt_REJECT
xt_tcpudp   20480 6
bridge      176128 1 br_netfilter
stp          16384 1 bridge
llc          16384 2 bridge,stp
eBtable filter 16384 0
ip6tables 36864 1 eBtable_filter
ip6tables 16384 0
ip6tables 32768 1 ip6table_filter
ipTable filter 16384 1
bpfILTER    24576 0
aufs        258048 0
ccm          20480 6
overlay     114688 0
bnep        24576 2
intel_rapl_msr 20480 0
intel_rapl_common 24576 1 intel_rapl_msr
```



```
> sudo rmmod cpu_grupo33.ko
```

Después ejecutaremos el siguiente comando, para mostrar el mensaje del modulo.

En este caso, de último se mostrarán el nombre del curso (es el mensaje que se muestra al deshabilitar nuestro módulo).

```
> dmesg
```

The image shows a Visual Studio Code editor window with a C program and its execution output.

File Explorer (Left):

- EXPLORER
- OPEN EDITORS
- Makefile
- cpu_201602434_20150... 7
- MODULOCPU
- cpu_201602434_201503984.ko...
- cpu_201602434_201503984.m...
- cpu_201602434_201503984.m...
- cpu_201602434_201503984.o...
- cpu_201602434_201503984.o...
- cpu_201602434_201503984.ko...
- cpu_201602434_201503984.mod
- cpu_201602434_201503984.o...
- cpu_201602434_201503984.mo...
- cpu_201602434_201503984.mo...
- Makefile
- Module.symvers
- modules.order
- OUTLINE
- TIMELINE

Editor (Center):

File: cpu_201602434_201503984.c

```

1 // ...
22 struct list_head *list;
23
24

```

Terminal (Right):

1: bash

```

[ 50.494932] wlp3s0: RX AssocResp from 00:21:94:12:42:78 (capab=0x411 status=0 aid=2)
[ 50.497456] wlp3s0: associated
[ 51.012353] IPV6: ADDRCONF(NETDEV_CHANGE): wlp3s0: link becomes ready
[ 52.044013] aufs 4.x-rcw-20180805
[ 52.913811] bpftr: Loaded bpftr_umh pid 1893
[ 52.914682] Started bpftr
[ 53.534332] bridge: filtering via arp/ip/ip6tables is no longer available by default. Update your scripts to load br_netf
ilter if you need this.
[ 53.537824] virbr0: port 1(virbr0-nic) entered blocking state
[ 53.537827] virbr0: port 1(virbr0-nic) entered disabled state
[ 53.538807] device virbr0-nic entered promiscuous mode
[ 54.230567] virbr0: port 1(virbr0-nic) entered blocking state
[ 54.230574] virbr0: port 1(virbr0-nic) entered listening state
[ 54.471411] virbr0: port 1(virbr0-nic) entered disabled state
[ 59.416956] kauditd_printk skb: 40 callbacks suppressed
[ 59.416957] audit: type=1400 audit(159204090.5752): apparmor="STATUS" operation="profile_load" profile="unconfined" na
me="docker-default" pid=2324 comm="apparmor parser"
[ 61.957624] Bridge firewalling registered
[ 62.161555] Initializing XFRM netlink socket
[ 166.739542] Bluetooth: RFCOMM TTY layer initialized
[ 166.739553] Bluetooth: RFCOMM socket layer initialized
[ 166.739563] Bluetooth: RFCOMM ver 1.11
[ 169.656938] rfkill: input handler disabled
[ 232.080482] wlp3s0: AP 00:21:94:12:42:78 changed bandwidth, new config is 2412 MHz, width 2 (2422/0 MHz)
[ 265.158197] audit: type=1408 audit(1592044255.97153): apparmor="DENIED" operation="capable" profile="/snap/core/9289/usr
/lib/snapd/snap" capability=4 capname="fsuid"
[ 2906.220902] perf: interrupt took too long (2620 > 2500), lowering kernel.perf.event_max_sample_rate to 76250
[ 3892.217258] cpu 201602434_201503984: loading out-of-tree module taints kernel.
[ 3892.217332] cpu 201602434_201503984: module verification failed: signature and/or required key missing - tainting kernel
[ 3892.218015] nombre1:Jeannira Del Rosario Sic Menéndez
[ 3892.218015] nombre2: Fernando Vidal Ruiz Piox
[ 4006.842836] Curso: Sistemas Operativos 1
feruiz@feruiz-Aspire-E5-473:~/Escritorio/ModuloCPU$

```

Front End

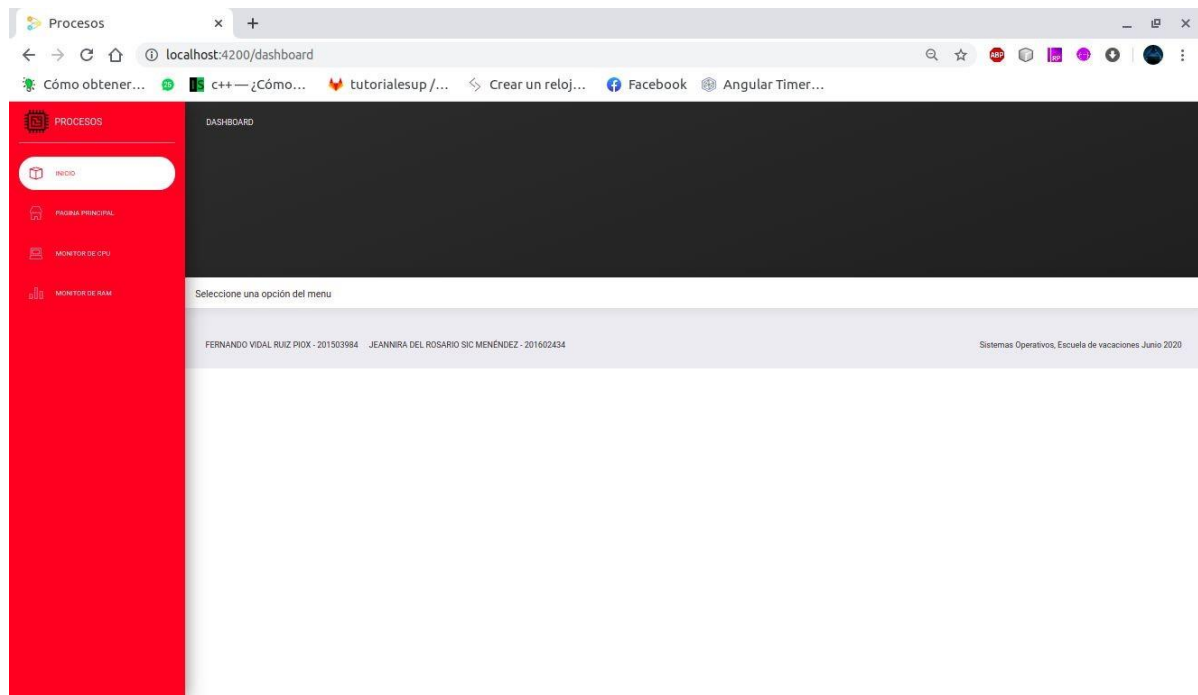
Aplicación WEB

La aplicación web permite visualizar gráficas dinámicas que muestren el uso del CPU y de la memoria RAM del servidor.

La aplicación web permite mostrar la información básica de los procesos que se ejecutan y permite terminar los procesos(kill) que se encuentran en ejecución.

Página de Inicio

Muestra la información general de los creadores del proyecto.



Página principal

Muestra tabulada todos los procesos que están siendo ejecutados en el servidor, así como un resumen general de los procesos. La información general que se muestra es la siguiente:

Procesos en ejecución	Número de procesos en estado de ejecución (running)
Procesos suspendidos	Número de procesos en estado de suspendido (sleeping)
Procesos detenidos	Número de procesos en estado de detenido (stoped)
Procesos Zombie	Número de procesos zombie
Total de procesos	Número total de procesos

Además muestra de manera tabulada la siguiente información de cada proceso:

PID	Identificador del proceso
Nombre	Nombre del proceso
Usuario	Usuario que ejecutó el proceso

Estado
%RAM

Estado en el que se encuentra el proceso
Porcentaje de utilización de RAM por el proceso

En esta sección, se encuentra un botón el cual desarrolla la función de terminar (kill) al proceso seleccionado en la tabla ya descrita.

The screenshot shows the 'Procesos' application interface. On the left is a red sidebar with navigation links: PROCESOS, REBO, PÁGINA PRINCIPAL (selected), MONITOR DE CPU, and MONITOR DE RAM. The main content area is titled 'Página principal' and contains a 'Resumen general' (General Summary) section with five statistics: Procesos en ejecución (1), Procesos suspendidos (226), Procesos detenidos (0), Procesos Zombie (0), and Total de procesos (284). Below this is a 'Listado de procesos' (Process List) section with a table showing the following data:

PID	Nombre	Usuario	Estado	%RAM	Acción
1	systemd	root	S	5.716816%	Kill
10	ksftirqd/0	root	S	0.000000%	Kill
1029	snapt	root	S	33.164278%	Kill
1031	thermald	root	S	4.736220%	Kill
1051	networkd-dispat	root	S	4.499151%	Kill

At the bottom of the table, there is a pagination control showing 'Items per page: 5' and '1 - 5 of 285'.

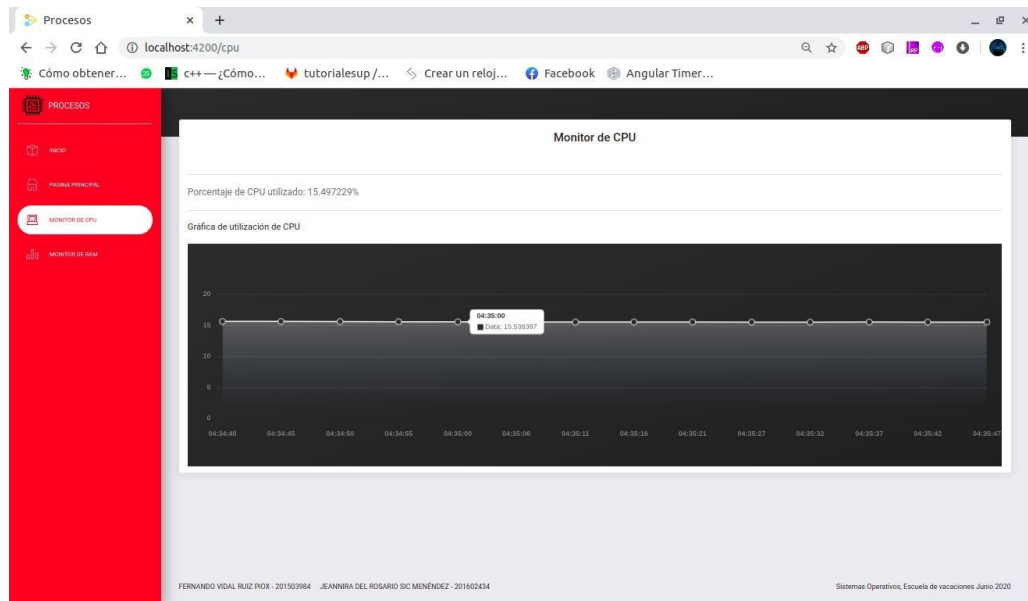
Se muestra un árbol de procesos dinámico, es decir, se muestra una lista de procesos que permita seleccionar uno y desplegar a todos sus hijos con PID y nombre.

The screenshot shows the 'Procesos' application interface with the 'Árbol de procesos' (Process Tree) section selected. The top part of the interface shows the details for PID 972, named 'avahi-daemon', owned by 'avahi', in state 'S', using 1.196896% of RAM, with a 'Kill' button. Below this is the 'Árbol de procesos' section, which displays a hierarchical tree of processes. The root node is 'PID: 1, Nombre: systemd'. It has several children, including 'PID: 2028, Nombre: dnsmasq', 'PID: 2080, Nombre: dockerd', 'PID: 3062, Nombre: rtkitd', 'PID: 2110, Nombre: kerneloops', 'PID: 2120, Nombre: kerneloops', 'PID: 2366, Nombre: upowerd', 'PID: 2609, Nombre: rtkit-daemon', 'PID: 2722, Nombre: bus-x11', 'PID: 2962, Nombre: htd', 'PID: 3002, Nombre: packagekitd', 'PID: 3182, Nombre: colord', 'PID: 326, Nombre: systemd-journal', and 'PID: 33, Nombre: home'. The 'PID: 2028, Nombre: dnsmasq' node is expanded, showing its own children: 'PID: 2715, Nombre: xdg-permission', 'PID: 2725, Nombre: dbus-portal', 'PID: 2830, Nombre: dnsmasq', 'PID: 3082, Nombre: (sd-pam)', 'PID: 3610, Nombre: dbus-daemon', 'PID: 3712, Nombre: at-spi-bus-launcher', 'PID: 3719, Nombre: at-spi2-regist', 'PID: 3744, Nombre: gvfud', 'PID: 3749, Nombre: gvfud-base', 'PID: 3790, Nombre: bus-portal', 'PID: 3791, Nombre: xdg-permission', 'PID: 3801, Nombre: gnome-shell-cal', 'PID: 3819, Nombre: gvfs-disk2-vo', and 'PID: 38, Nombre: evolution-rt'. The 'PID: 3712, Nombre: at-spi-bus-launcher' node is further expanded, showing 'PID: 3717, Nombre: dbus-daemon', 'PID: 4025, Nombre: gvfud-trash', 'PID: 6471, Nombre: gvfud-network', and 'PID: 8588, Nombre: gvfud-dread'. At the bottom of the page, there is a footer with the text 'FERNANDO VIDAL RUIZ PLOX - 201503984 JEANNIRA DEL ROSARIO SIC MENÉNDEZ - 201602434' and 'Sistemas Operativos, Escuela de vacaciones Junio 2020'.

Monitor de CPU

El monitor de CPU debe muestra la información del consumo de CPU del servidor, en el cual se visualiza la siguiente información:

- Porcentaje de CPU utilizado
- Gráfica de utilización de CPU en tiempo real



Monitor de RAM

El monitor de memoria RAM es similar al de CPU, muestra la información del consumo de RAM del servidor, en el cual se visualiza la siguiente información:

- Total de memoria RAM del servidor (en MB)
- Total de memoria RAM consumida (en MB)
- Porcentaje de consumo de RAM

