OS 2021
Homework1

(Due day 2021/11/08 23:59:59)

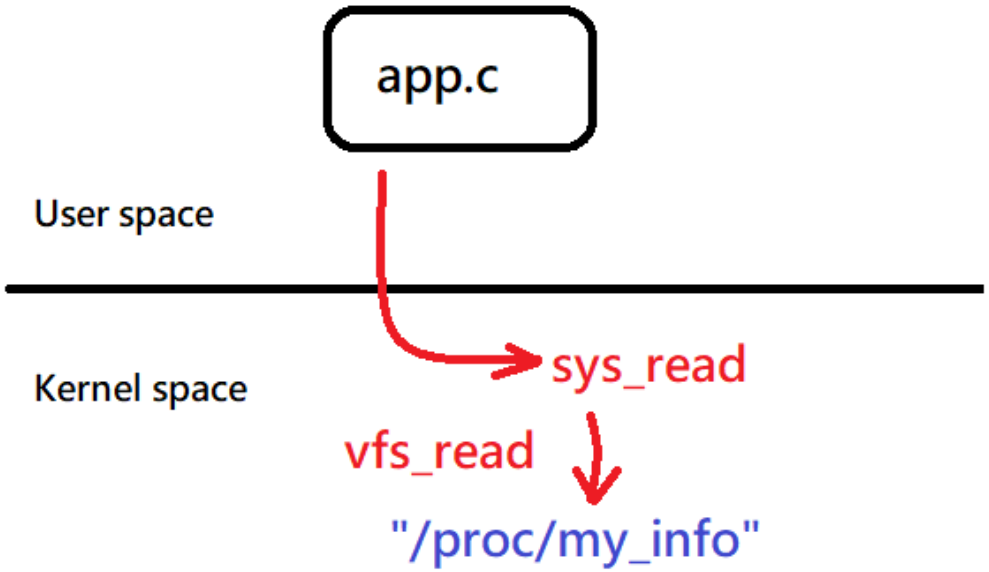
TA : 葉怡君

E-mail : P76094274@gs.ncku.edu.tw

Objectives

- Understand how to program a kernel module.
- Understand and write your own proc interface.

Overview



- **Create a read and write proc entry in the kernel version 4.15.**
- **app call `read()` operations to read the proc entry**
- **app displays the message**

Requirement – kernel module

- Create your own proc file to show some information.
  - Module name：my_info
  - The file /proc/my_info is created when the module is loaded.
- Grab system information and record them to proc file.
- Cannot read existing proc files directly.
  - Grab the information by yourself.

Example：Linux version

/fs/proc/version.c

```
 9  static int version_proc_show(struct seq_file *m, void *v)
10  {
11          seq_printf(m, linux_proc_banner,
12                  utsname()->sysname,
13                  utsname()->release,
14                  utsname()->version);
15          return 0;
16  }
17
```

Example：Linux version

/init/version.c

```
25  struct uts_namespace init_uts_ns = {
26          .kref = KREF_INIT(2),
27          .name = {
28                  .sysname        = UTS_SYSNAME,
29                  .nodename       = UTS_NODENAME,
30                  .release        = UTS_RELEASE,
31                  .version        = UTS_VERSION,
32                  .machine        = UTS_MACHINE,
33                  .domainname     = UTS_DOMAINNAME,
34          },
35          .user_ns = &init_user_ns,
36          .ns.inum = PROC_UTS_INIT_INO,
37  #ifdef CONFIG_UTS_NS
38          .ns.ops = &utsns_operations,
39  #endif
40  };
41  EXPORT_SYMBOL_GPL(init_uts_ns);
42
43  /* FIXED STRINGS! Don't touch! */
44  const char linux_banner[] =
45          "Linux version " UTS_RELEASE " (" LINUX_COMPILE_BY "@"
46          LINUX_COMPILE_HOST ") (" LINUX_COMPILER ") " UTS_VERSION "\n";
47
48  const char linux_proc_banner[] =
49          "%s version %s"
50          " (" LINUX_COMPILE_BY "@" LINUX_COMPILE_HOST ")"
51          " (" LINUX_COMPILER ") %s\n";
```

generated/utsrelease.h

Include <generated/utsrelease.h>

```
seq_puts(m, "\n=============Version=============\n");
seq_printf(m, "Linux version %s\n", UTS_RELEASE);
```

Requirement – proc file information

- Version：
  - Linux version
- CPU：
  - processor、model name、physical-id、core-id、cache size、clflush-size、cache-alignment、address-sizes
- Memory：
  - Memtotal、Memfree、Buffers、Activate、Inactivate、Shmem、Dirty、Writeback、KernelStack、PageTables
- Time：
  - Uptime、Idletime

Requirement – user space application

- Read computer information under your /proc file.
  - Cannot use the library or system call (e.g. sysinfo()) to grab the information.
- Can choose what information to display.
  - Version
  - CPU
  - Memory
  - Time
  - All
- Implement interface follow page 9.

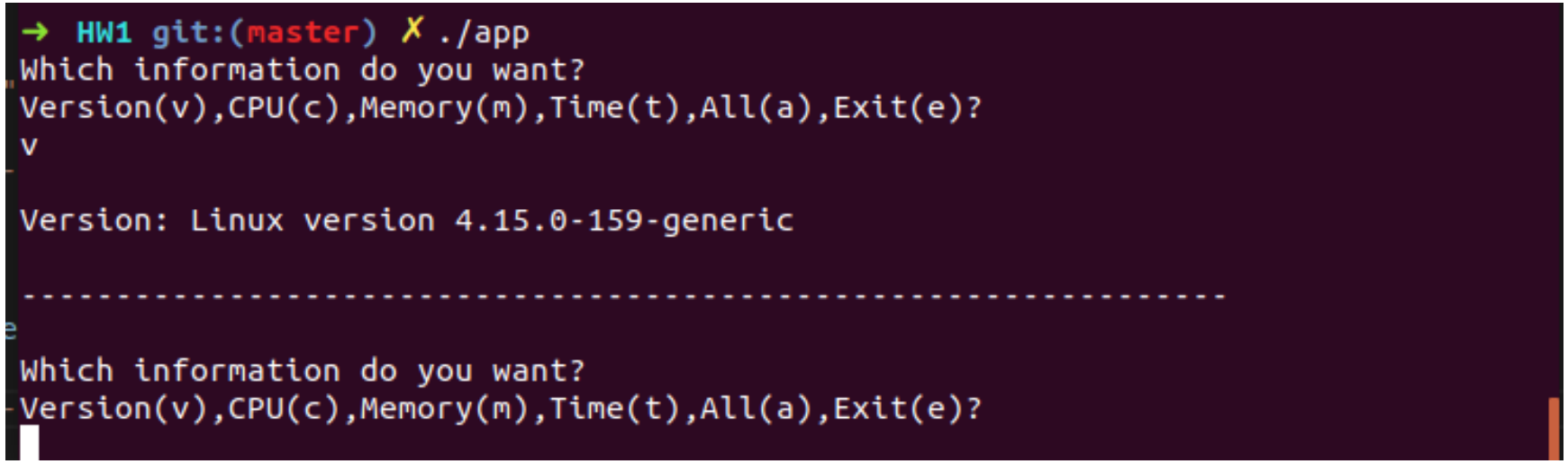
Requirement – user space application interface

- **"Which information do you want?"**
  "Version(v),CPU(c),Memory(m),Time(t),All(a),Exit(e)?"
- Input your choice.
- Output data format：
  - **"Version："**
  - **"CPU information：\n"**
  - **"Memory information：\n"**
  - **"Time information：\n"**
- Choose "All" will show your proc file.

Example-proc file

```
➜ module cat /proc/my_info

=============Version=============
Linux version 4.15.0-159-generic

=============CPU================
processor       : 0
model name      : Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz
physical id     : 0
core id         : 0
cpu cores       : 1
cache size      : 6144 KB
clflush size    : 64
cache_alignment : 64
address sizes   : 39 bits physical, 48 bits virtual


============Memory==============
MemTotal        : 1008732 kB
MemFree         : 112932 kB
Buffers         : 7800 kB
Active          : 366292 kB
Inactive        : 362228 kB
Shmem           : 59676 kB
Dirty           : 520 kB
Writeback       : 0 kB
KernelStack     : 45232 kB
PageTables      : 46164 kB

=============Time================
Uptime          : 1835.34 (s)
Idletime        : 1482.74 (s)
```

Example-Application



Example-Application

```
➜  HW1 git:(master) ✗ ./app
Which information do you want?
Version(v),CPU(c),Memory(m),Time(t),All(a),Exit(e)?
v

Version: Linux version 4.15.0-159-generic

------------------------------------------------------------

Which information do you want?
Version(v),CPU(c),Memory(m),Time(t),All(a),Exit(e)?
c

Cpu information:
processor       : 0
model name      : Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz
physical id     : 0
core id         : 0
cpu cores       : 1
cache size      : 6144 KB
clflush size    : 64
cache_alignment : 64
address sizes   : 39 bits physical, 48 bits virtual

------------------------------------------------------------

Which information do you want?
Version(v),CPU(c),Memory(m),Time(t),All(a),Exit(e)?
```

Example-Application

```
Which information do you want?
Version(v),CPU(c),Memory(m),Time(t),All(a),Exit(e)?
a


=============Version=============
Linux version 4.15.0-159-generic

=============CPU================
processor       : 0
model name      : Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz
physical id     : 0
core id         : 0
cpu cores       : 1
cache size      : 6144 KB
clflush size    : 64
cache_alignment : 64
address sizes   : 39 bits physical, 48 bits virtual


============Memory==============
MemTotal        : 1008732 kB
MemFree         : 69036 kB
Buffers         : 12168 kB
Active          : 384904 kB
Inactive        : 367012 kB
Shmem           : 41100 kB
Dirty           : 172 kB
Writeback       : 0 kB
KernelStack     : 50752 kB
PageTables      : 54064 kB

=============Time================
Uptime          : 498409.79 (s)
Idletime        : 493537.86 (s)



----------------------------------------------------------------

Which information do you want?
Version(v),CPU(c),Memory(m),Time(t),All(a),Exit(e)?
e
➜  HW1 git:(master) ✗
```

Github classroom

- Click here to start your assignment.
- **Due day 2021/11/08 23:59:59**

Reference

- The Linux Kernel Module Programming Guide
- The Linux Kernel documentation
- Linux Kernel Development (chapter 3)
- Linux Device Drivers (chapter 2)
- Proc filesystem
- Linux source code