Lab 5

Members: Fangwen Liao (3439869) | Yijin Wang (3476217)

Exercise 1

The Linux kernel enforces the following policy: a process is not allowed to try to acquire a semaphore while it is already holding a spin lock.

(a) Explain why this policy has to be enforced.

When a process hold a spin lock, it will loop until it gets the resource, it should not wait for a long time. When holding a semaphore, a process can go to sleep, so a process can not go to sleep while holding a spin lock, or when it sleeps it can not release the spin lock.

(b) What can happen when the policy is not enforced? The process holding a spin lock and a semaphore can go to sleep and neverrelease the spin lock, this may causing deadlock, or it should go to sleep, but spinning and waste cpu cycles. If a process grabs a spinlock and goes to sleep before releasing it. A second process (or an interrupt handler) that to grab the spinlock will busy wait. On an uniprocessor machine the second process will lock the CPU not allowing the first process to wake up and release the spinlock so the second process can continue, it is basically a deadlock.

Exercise 2

Describe the problem that occurs when using spinlocks on a single processor system. Consider both the case of

(a) a preemptible kernel and

When a process holding a spinlock is preempted, it doesn't release the lock, if the preempting process need the resource, it will never get it, thus forming a deadlock. Spinlock automatically disables preemption, which avoids deadlock caused by interrupts. when data is shared with interrupt handler, before holding spinlock we must disable interrupts.

(b) a non preemptible kernel.

In a non-preemptible kernel, a process will release the resource only when it finishes using it and will not causing a dead lock for the next process.

Exercise 3

Explain what concurrency issues can arise when the up() (increment counter) and down() (decrement counter) functions of a semaphore are not executed atomically. Illustrate these issues with an example using 2 threads.

when Thread0 aquire the semaphore, the count should be decreased from 1 to zero, but as down() not atomically runned, the count stays 1, thus Thread1 can also get the resource, thus causing data inconsistancy. When Thread0 finishes using resrouce and release the semaphore, as up() not excuted atomically, Thread1 will never be able to get the resource.

Download OS_Lab5.zip and unzip it. Open the kernel directory. It is recommended to make a linked clone of your virtual machine before continuing.

Exercise 4

*Inspect the .*sh scripts and download + compile the Linux kernel using these scripts.*

(a) Show the output of uname a before installation.

The provided sh files have problems when running, to solve this problem, we copy them and run them inside the virtual machinei.

```
fangwenliao@debian:~/Downloads/OS_Lab5/kernel$ uname -a
Linux debian 4.9.0-16-686 #1 SMP Debian 4.9.272-2 (2021-07-19) i686 GNU/Linux
```

(b) Install the four Debian packages that are created once the compilation of the kernel completes successfully.

To run the compile.sh, we first need to apt install time or it will have errors when running the script, or we can just delete the time, since only the make command really matters. After the compile.sh is done, change to root user and use the -i option of dkpg to install the pakage.

```
root@debian:/home/yijinwang/Documents/OS_Lab5/kernel# dpkg -i linux-headers-4.19.152_4.19.152-1_i386.deb
Selecting previously unselected package linux-headers-4.19.152.
(Reading database ... 166573 files and directories currently installed.)
Preparing to unpack linux-headers-4.19.152_4.19.152-1_i386.deb ...
Unpacking linux-headers-4.19.152 (4.19.152-1) ...
Setting up linux-headers-4.19.152 (4.19.152-1) ...
```

```
root@debian:/home/yijinwang/Documents/OS_Lab5/kernel# dpkg -i linux-libc-dev_4.19.152-1_i386.deb
(Reading database ... 190359 files and directories currently installed.)
Preparing to unpack linux-libc-dev_4.19.152-1_i386.deb ...
Unpacking linux-libc-dev (4.19.152-1) over (4.9.290-1) ...
Setting up linux-libc-dev (4.19.152-1) ...
```

```
root@debian:/home/yijinwang/Documents/OS_Lab5/kernel# dpkg -i linux-image-4.19.152_4.19.152-1_i386.deb
Selecting previously unselected package linux-image-4.19.152.
(Reading database ... 190543 files and directories currently installed.)
Preparing to unpack linux-image-4.19.152_4.19.152-1_i386.deb ...
Unpacking linux-image-4.19.152 (4.19.152-1) ...
Setting up linux-image-4.19.152 (4.19.152-1) ...
update-initramfs: Generating /boot/initrd.img-4.19.152
```

```
Generating grub configuration file ...
Found background image: /usr/share/images/desktop-base/desktop-grub.png
Found linux image: /boot/vmlinuz-4.19.152
Found initrd image: /boot/initrd.img-4.19.152
Found linux image: /boot/vmlinuz-4.9.0-17-686
Found initrd image: /boot/initrd.img-4.9.0-17-686
Found linux image: /boot/vmlinuz-4.9.0-16-686
Found initrd image: /boot/initrd.img-4.9.0-16-686
Found linux image: /boot/vmlinuz-4.9.0-7-686
Found initrd image: /boot/initrd.img-4.9.0-7-686
done
```

```
root@debian:/home/yijinwang/Documents/OS_Lab5/kernel# dpkg -i linux-image-
4.19.152-dbg_4.19.152-1_i386.deb
Selecting previously unselected package linux-image-4.19.152-dbg.
(Reading database ... 190735 files and directories currently installed.)
Preparing to unpack linux-image-4.19.152-dbg_4.19.152-1_i386.deb ...
Unpacking linux-image-4.19.152-dbg (4.19.152-1) ...
Setting up linux-image-4.19.152-dbg (4.19.152-1) ...
```

*(c) Reboot into the newly installed kernel. Show the output of **uname a.***

```
fangwenliao@debian:~$ uname -a
Linux debian 4.19.152 #1 SMP Sat Jan 8 08:30:56 CET 2022 i686 GNU/Linux
```

Exercise 5

Implement your own system call that (1) takes one argument (a nice value), (2) prints the current process' PID and nice value, (3) changes the current process' nice value to the nice value that is given as a parameter of the system call, and (4) finally returns the process's PID to user space.

(a) Implement your system call in sys.c using the appropriate macro,

our code is as below, the task_tgid_vnr, find_task_by_vpid, task_nice and set_user_nice are all used in sys.c before, it is fine to use them here, we put them between line 2579 and line 2591, note it can not be put inside #ifdef, otherwise it will get errors while compiling.

```
SYSCALL_DEFINE1(newcall, int, nice_value)
{
        int pid;
        struct task_struct *p;
        int ni;

        pid = task_tgid_vnr(current);
        p = find_task_by_vpid(pid);
        ni = task_nice(p);
        printk("PID: %d, Nice Value:%d.\n", pid, ni);
        set_user_nice(p, nice_value);
        return pid;
}
```

(b) Add an entry for the new system call to the syscall_32.tbl for your appropriate architecture,

we enter the following code in syscall_32.tbl

```
387     i386    newcall                 sys_newcall
__ia32_sys_newcall
```

(c) Compile, install, and boot the kernel containing your system call,

repeat the process in 4

(d) Implement a C application (in user space) that calls your newly implemented system call using the syscall function,

our code is like below:

```
#include <stdio.h>

int main(void)
{
        int pid;

        pid = syscall(387, 11);
        printf("Current PID: %d", pid);
        while(1)
        {

        }

        return pid;
}
```

(e) Run your application and show that the system call is made by inspecting the strace.

```
fangwenliao@debian:~/Downloads/kernel$ gcc test.c
test.c: In function ‘main’:
test.c:7:8: warning: implicit declaration of function ‘syscall’ [-Wimplicit-function-declaration]
  pid = syscall(387, 11);
fangwenliao@debian:~/Downloads/kernel$ strace ./a.out
execve("./a.out", ["./a.out"], [/* 48 vars */]) = 0
brk(NULL)                               = 0x10cf000
```

```
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
mmap2(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0xb7ed9000
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=100353, ...}) = 0
mmap2(NULL, 100353, PROT_READ, MAP_PRIVATE, 3, 0) = 0xb7ec0000
close(3)                                = 0
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
open("/lib/i386-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\1\1\1\3\0\0\0\0\0\0\0\0\3\0\3\0\1\0\0\0\204\1\0004\0\0\0"...,
512) = 512
fstat64(3, {st_mode=S_IFREG|0755, st_size=1791908, ...}) = 0
mmap2(NULL, 1800700, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0xb7d08000
mprotect(0xb7eb9000, 4096, PROT_NONE)   = 0
mmap2(0xb7eba000, 12288, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1b1000) = 0xb7eba000
mmap2(0xb7ebd000, 10748, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0xb7ebd000
close(3)                                = 0
set_thread_area({entry_number:-1, base_addr:0xb7eda100, limit:1048575,
seg_32bit:1, contents:0, read_exec_only:0, limit_in_pages:1, seg_not_present:0,
useable:1}) = 0 (entry_number:6)
mprotect(0xb7eba000, 8192, PROT_READ)   = 0
mprotect(0x453000, 4096, PROT_READ)     = 0
mprotect(0xb7f03000, 4096, PROT_READ)   = 0
munmap(0xb7ec0000, 100353)              = 0
syscall_387(0xb, 0x1, 0x4525e7, 0x1, 0xbfefcc24, 0xbfefcc2c) = 0x6ab
fstat64(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(136, 0), ...}) = 0
brk(NULL)                               = 0x10cf000
brk(0x10f0000)                          = 0x10f0000
^Z
[1]+  Stopped                 strace ./a.out
```

(f) Show that the nice value of the process has changed. TIPS: use the current macro, and set_user_nice function

we use the ps command to check the nice value of a.out, as is shown the nice value has changed to the value we wanted.

```
fangwenliao@debian:~/Downloads/kernel$ ps -al
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S   117   761   757  0  80   0 - 19524 -      tty1     00:00:00 gnome-session-
0 S   117   769   761  1  80   0 - 226673 -     tty1     00:00:03 gnome-shell
0 S   117   791   769  0  80   0 - 36889 -      tty1     00:00:00 Xwayland
0 S   117   827   761  0  80   0 - 129986 -     tty1     00:00:00 gnome-settings
4 S  1000  1146  1144  1  80   0 - 45691 epoll_ tty2     00:00:02 Xorg
0 S  1000  1160  1144  0  80   0 - 19557 poll_s tty2     00:00:00 gnome-session-
0 S  1000  1290  1160  4  80   0 - 237140 poll_s tty2    00:00:09 gnome-shell
0 S  1000  1392  1160  0  80   0 - 105763 poll_s tty2    00:00:00 gnome-settings
0 S  1000  1412  1160  6  99  19 - 49561 poll_s tty2     00:00:12 tracker-extrac
```

```
  0 S  1000  1415  1160  0  80   0 - 40516 poll_s tty2     00:00:00 gnome-software
  0 S  1000  1416  1160 53  99  19 - 34674 poll_s tty2     00:01:43 tracker-miner-
  0 S  1000  1417  1160  0  99   - - 15567 poll_s tty2     00:00:00 tracker-miner-
  0 S  1000  1420     1  0  80   0 - 18219 poll_s tty2     00:00:00 gsd-printer
  0 S  1000  1421  1160  0  80   0 - 59929 poll_s tty2     00:00:00 evolution-alar
  0 S  1000  1424  1160  0  99   - - 21444 poll_s tty2     00:00:00 tracker-miner-
  0 T  1000  1705  1630  0  80   0 -   709 signal pts/0    00:00:00 strace
  0 t  1000  1707  1705 42  91  11 -   553 ptrace pts/0    00:00:07 a.out
  0 R  1000  1708  1630  0  80   0 -  1851 -      pts/0    00:00:00 ps
```