HCMC University Of Technology
Faculty of Computer Science & Engineering

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

OPERATING SYSTEMS

Assignment #1 - System Call

Instructor: Nguyen Minh Tri
Student: 160852 - Huynh Sam Ha

Ho Chi Minh City, 3/2018

Mục lục

1 Prepare Linux Kenel **2**
- 1.1 QUESTION: Why we need to install kernel-package? . . . 2
- 1.2 QUESTION: Why we have to use another kernel source from the server such as http://www.kernel.org, can we compile the original kernel (the local kernel on the running OS) directly? . . . 2

2 System Call - procsched **2**
- 2.1 QUESTION: What is the meaning of other parts, i.e. i386, procsched, and sys procsched? . . 2
- 2.2 QUESTION: What is the meaning of each line added in file include/linux/syscalls.h? . . . 2

3 Compiling Linux Kernel **2**
- 3.1 QUESTION: What is the meaning of these two stages, namely "make" and "make modules"? . 2
- 3.2 QUESTION: Why this program could indicate whether our system works or not? . . . 2

4 Wrapper **3**
- 4.1 QUESTION: Why we have to re-define proc segs struct while we have already defined it inside the kernel? . . . 3
- 4.2 QUESTION: Why root privilege (e.g. adding sudo before the cp command) is required to copy the header file to /usr/include? . . . 3
- 4.3 QUESTION: Why we must put -share and -fpic option into gcc command? . . . 3

1 Prepare Linux Kenel

1.1 QUESTION: Why we need to install kernel-package?

kernel-package is utility for building Linux kernel related Debian packages. The package automate the routine steps required to compile and install a custom kernel.
For more advantages, see http://man.he.net/man5/kernel-package

1.2 QUESTION: Why we have to use another kernel source from the server such as http://www.kernel.org, can we compile the original kernel (the local kernel on the running OS) directly?

2 System Call - procsched

2.1 QUESTION: What is the meaning of other parts, i.e. i386, procsched, and sys procsched?

Each system call is declared in one row with following information: number, ABI, name, entry point and compat entry point.

- <**abi**>: Application Binary Interface, i386 for system 32-bit and

2.2 QUESTION: What is the meaning of each line added in file include/linux/syscalls.h?

- struct proc_segs;

  declare one struct in the header file that is used in function sys_procsched.

- asmlinkage long sys_procsched(int pid, struct proc_segs * info);

  declares one asmlinkage function. The keyword asmlinkage tells compiler to look on the CPU stack for the function parameters, instead of registers. The interesting part is why this is necessary. System calls are services that userspace can call to request the kernel to perform something for them (and therefore execute in kernel space). These functions are quite strange that you cannot expect them to behave like normal functions, where parameters are typically passed by writing to the program stack, but instead they are written to registers.

3 Compiling Linux Kernel

3.1 QUESTION: What is the meaning of these two stages, namely "make" and "make modules"?

- make: to compile the main kernel and create *vmlinuz*.

  + *vmlinuz*: is the name of the Linux kernel executable. That is a compressed Linux kernel, and it is bootable (capable of loading the operating system into memory).

- make modules: to compile the kernel modules.

3.2 QUESTION: Why this program could indicate whether our system works or not?

Because we call the system call through library $< sys/syscall.h >$, which calls $syscall([number\_32], [pid], info);$ with $number\_32$ is the number of procsched system call in the file $syscall32.tlb$.
If our kernel not working, that means get fail after compiling or building kernel, the syscall with $number\_32$ is also not working. Opposite, the syscall will return params to $info$ as our kernel implementation.
Therefore, we can detect if our system works or not via the program.

4 Wrapper

4.1 QUESTION: Why we have to re-define proc segs struct while we have already defined it inside the kernel?

The struct $proc\_segs$ we have already defined is only used in the kernel (in kernelspace). We must declare again the struct $proc\_segs$ as the utility module for user to use easier (in userspace).
We can declare this struct is not similar to struct in kernel we defined. But we should use the returned params from syscall correctly with their meaning.

4.2 QUESTION: Why root privilege (e.g. adding sudo before the cp command) is required to copy the header file to /usr/include?

$sudo$ (superuser do) is used for root account permission.
$/usr/$ is owned by the root account so to write files in there you need to write them as root. Therefore, we should add sudo to do command as root account.

Therefore, we must have $sudo$ before copy file to $/usr/$.

4.3 QUESTION: Why we must put -share and -fpic option into gcc command?

The meaning each option gcc is:

- -shared: to create a shared library.
- -fpic:to generate position-independent code (PIC) suitable for use in a shared library, if supported for the target machine.

For more detail, see here: https://www.cs.swarthmore.edu/ newhall/unixhelp/howto_C_libraries.html
We are trying to create a Shared Object File (file .so - $libprocsched.so$) that can be linked into other programs that use owr library. Therefore, we must add flag $-shared$ to create file .so above. And we also add flag $-fpic$ as the meaning of this option.