OSPJ1 report

Team04 B03202040 吳俊儒 B04901009 林孟瑾

1. Implementation

After building the Linux kernel:

(1) Add system calls to syscall_table_32.S

```
.long sys_hello
.long sys_multiply
.long sys_min
```

(2) Define macros in unistd_32.h

```
#define __NR_hello              337
#define __NR_multiply           338
#define __NR_min                339

#ifdef __KERNEL__

#define NR_syscalls 340
```

(3) Define prototypes of system calls in syscalls.h

```
asmlinkage long sys_hello(void);

asmlinkage long sys_multiply(long,long);

asmlinkage long sys_min(long,long);
#endif /* CONFIG_X86_32 */
#endif /* _ASM_X86_SYSCALLS_H */
```

(3) Functions of system calls

```
#include <linux/kernel.h>
#include <linux/linkage.h>

asmlinkage long sys_hello(void)
{
        printk("HELLO SYSTEM CALL b03202040 b04901009\n");
        return 0;
}
```

```
#include <linux/kernel.h>
#include <linux/linkage.h>

asmlinkage long sys_multiply(long a,long b)
{
        return (a*b);
}
```

```
#include <linux/kernel.h>
#include <linux/linkage.h>

asmlinkage long sys_min(long a,long b)
{
        if (a < b){return a;}
        else{return b;}
}
```

(4) Modify Makefile to merge system calls into the kernel

```
obj-y     = sched.o fork.o exec_domain.o panic.o printk.o \
            cpu.o exit.o itimer.o time.o softirq.o resource.o \
            sysctl.o capability.o ptrace.o timer.o user.o \
            signal.o sys.o kmod.o workqueue.o pid.o \
            rcupdate.o extable.o params.o posix-timers.o \
            kthread.o wait.o kfifo.o sys_ni.o posix-cpu-timers.o mutex.o \
            hrtimer.o rwsem.o nsproxy.o srcu.o semaphore.o \
            notifier.o ksysfs.o pm_qos_params.o sched_clock.o cred.o \
            async.o hello.o multiply.o min.o
```

2. Difficulties

- Lack of memory in virtual machine:
  When building the kernel, there was an error message about running out of space. I solved this problem by reinstalling the virtual machine with bigger size of hard drive storage (8GB -> 15GB).

3. Results

(1) Test program

```c
#include <sys/syscall.h>
#include <unistd.h>
#include <stdio.h>

int main()
{

        long a = 123;
        long b = 100;
        syscall(337);
        printf("first number is : %ld\n",a);
        printf("second number is : %ld\n",b);
        printf("multiply of two numbers : %ld\n",syscall(338,a,b));
        printf("minimum of two numbers : %ld\n",syscall(339,a,b));
        return 0;

}
```

(2)Result

```
muachilin@muachilin-VirtualBox:~$ gcc test.c -o test
muachilin@muachilin-VirtualBox:~$ ./test
first number is : 123
second number is : 100
multiply of two numbers : 12300
minimum of two numbers : 100
muachilin@muachilin-VirtualBox:~$
```

```
[ 8761.649092] HELLO SYSTEM CALL b03202040 b04901009
```