Add a New System Call with args to copy file in kernel

Target

1. Add a new system call with arguments into the linux kernel
2. The new system call will return all processes information to user mode
    1. **实现内核中文件拷贝、并返回用户态** (见pdf文档)

Tools

Install GCC Software Colletion

```
sudo apt-get install build-essential
```

How to use GCC

- gcc and make

How to do

step 1:

in include/linux/syscalls.h

add the defintion of the new system call:

```
asmlinkage long sys_alcall(int cmd, char* buf);
```

step 2:

in kernel/sys.c

Please modify the implementation, because it is just a framework.

```
SYSCALL_DEFINE2(alcall,int,cmd,char*,buf)
{
    struct task_struct *p;
    printk("Hello new system call alcall (%d,%x)!\n",cmd,buf);
    print("%-20s %-6s %-6s\n","Name","Pid","Stat");
    for (p = &init_task; (p = next_task(p)) != &init_task;)
         printk("%-20s %-6d %-6ld\n",p->comm,p->pid,p->state);
    return 0;
}
```

functions in kernel:

```
copy_to_user
copy_from_user
sprintf
strcpy
strcat
...
```

step 3:

in arch/x86/entry/syscalls/syscall_64.tbl

add:

```
4??     common  alcall          __x64_sys_alcall
```

step 4:

re-configure the kernel

download the configure file in the current project: （optional）linux_config

in the directory of the kernel source code:

```
cp linux_config_ref20201203 .config
make oldconfig
make gconfig
```

recompile the kernel

```
make clean
make -j5
sudo  make  modules_install
sudo make install
```

reboot with the new kernel

reboot

step 5:

write a test program in user mode

see the previous lab.