**CS202 Lab1 Report**

Song Bai ?????

Yiqing Liu 862188893

1. **The list of all files modified**

**usys.S**

**syscall.h**

**syscall.c**

**sysproc.c**

**proc.c**

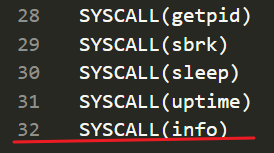
**defs.h**

**user.h**

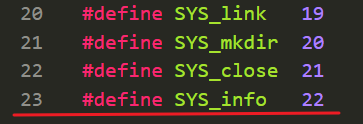
**info.c**

**Makefile**

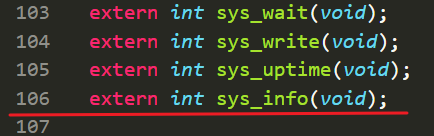
1. **A detailed explanation on what changes you have made and screenshots showing your work and results**
2. **System modification**
3. **In the file ‘usys.S’, the newly created system call ‘info’ is declared so that processes can recognize this function call as a system call.**

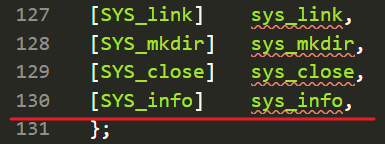


1. **In the file ‘syscall.h’, the system call number for the newly created system call ‘info’ is defined.**

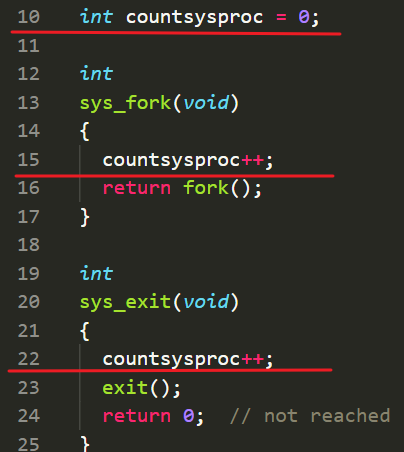


1. **In the file ‘syscall.c’, the system call handler of ‘info’ is added into the system call table.**

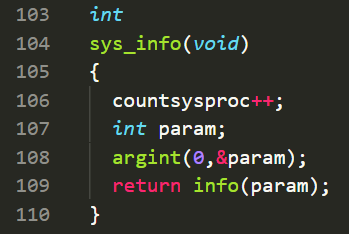




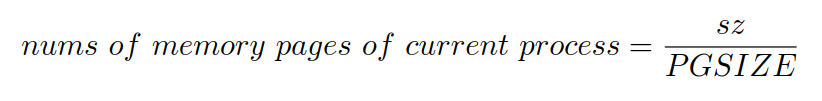
1. **In the file ‘sysproc.c’, a global counter named ‘countsysproc’ is incremented in each function inside ‘sysproc.c’ to count the total number of system calls that the current process has made.**

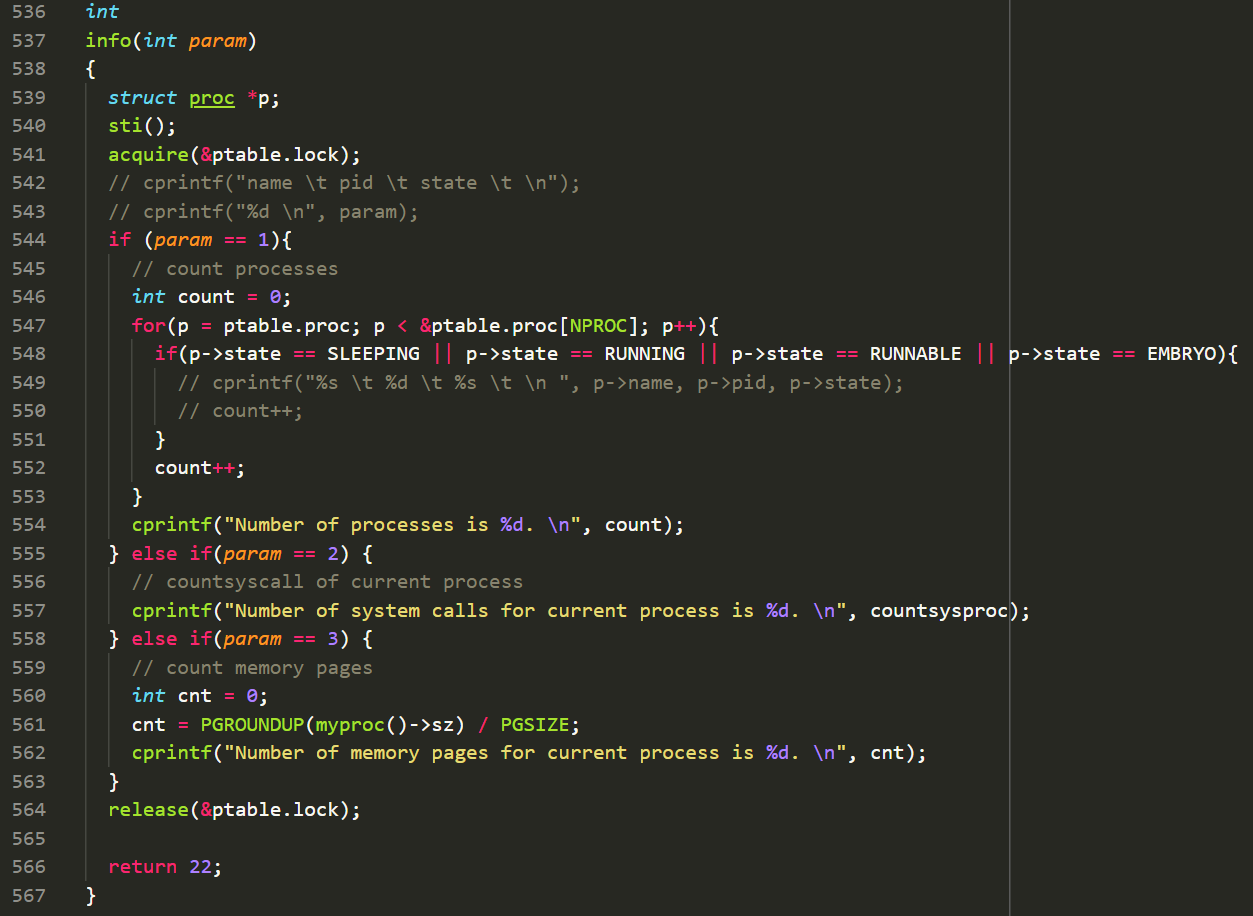


**And the body of the info system call function is defined as function ‘sys\_info’. It calls the function ‘info’ defined in the file ‘proc.c’ with the parameter user entered got by function ‘argint()’.**

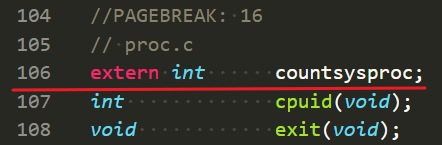


1. **In the file ‘proc.c’, the actually logic of ‘info’ system call is achieved in function ‘info’. The function switch between 3 different cases to achieve the function required by this lab. If the passed-in parameter is ‘1’, the function look through ptable and count all processes whose state is not ‘UNUSED’. If the passed-in parameter is ‘2’, the total number of system calls that the current process has made is printed. The information is stored in variable ‘countsysproc’, which is declared as a global counter in ‘defs.h’ and incremented** **in each function inside ‘sysproc.c’. If the passed-in parameter is ‘3’, the number of memory pages the current process is using is calculated and printed. As the following formula in image shows, the desired value can be calculated by dividing the total size of memory space of the current process with the size of each memory page in the system. The former can be acquired in the ‘sz’ field of ptable, and the latter can be acquired by the parameter PGSIZE.**

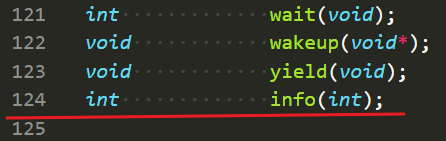




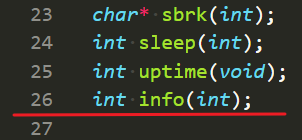
1. **In the file ‘defs.h’, a global counter variable ‘countsysproc’ is defined to count the total number of system calls that the current process has made.**



**The caller function ‘info’ is declared so that it can be called by other files in XV6 operating system.**



1. **In the file ‘user.h’, the ‘info’ function is added so that this function can be called by the user.**

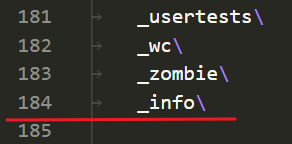


1. **System test**

**In order to test the functionality of the ‘info’ system call, a user level program named ‘info.c’ is created. Within this file, the newly added system call ‘info’ to XV6 is called.**

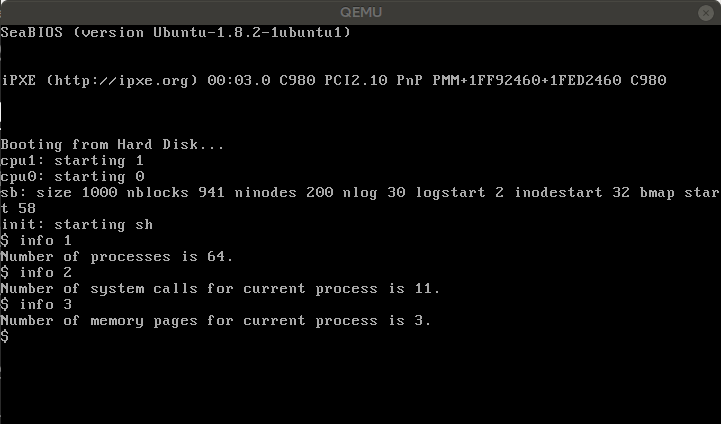


**The user level program ‘info.c’ is added into the Makefile so that it can work in XV6.**



1. **Test result**

**In our test, the info system call works as we expect.**



1. **A detailed description of XV6 source code (including your modifications) about how the info system call is processed, from the user-level program into the kernel code, and then back into the user-level program.**