Project 2.B Detailed Solution

1. Configuration: I have configured xv6-public on my Ubuntu Linux, but I do not have build-essentials and gawk, so install it using command

sudo apt-get install **build-essential** gawk qemu expect

1. Kernel Programming: I mainly worked on 8 files step by step
   1. syscall.h
      1. #define SYS\_getreadcount 22
   2. defs.h
      1. int getreadcount(void); //this is a declaration
   3. user.h
      1. int getreadcount(void);//this direction will be directly interfaced with kernel users
   4. sysproc.c

int sys\_getreadcount(void)

{

return getreadcount();

}//it is directly interfaced to the system calling action

* 1. syscall.c

extern int sys\_getreadcount(void); //linked to sysproc.c, thus enabling calling system process getreadcount () when needed

[SYS\_getreadcount] sys\_getreadcount, //Add the system call of getreadcount() to the system call table.

* 1. Usys.S

SYSCALL(getreadcount) //I imagine it like bind getreadcount to system call table, too

* 1. syscall.c:

extern int readcalledcount ;

int getreadcount()

{

return readcalledcount;

}

The most important part is the ***extern int readcalledcount*** ; readcalledcount is defined in the same file of int sysread(void), which will be called once a user call read() system call indirectly or directly, so I define a global variable ***readcalledcount*** to record the call of this system call. int getreadcount() is very easy, it is just returning a value.

* 1. create getreadcount.c

This will be the kernel level application, and the implementation is easy.

int main(int argc, char \*argv[])

{

getreadcount(); //just call the system call

exit();

}

* 1. make some change in the make file

UPROGS = \_getreadcount\

EXTRA=getreadcount.c