**Project 2B Report: xv6 System Call**

*Zhenxiong Han, zh5267, n.han@foxmail.com*

*Date: Feb 24, 2020*

Acknowledgement: I acknowledge that I am submitting this project late and will get points off due to late submission.

To implement the system call getreadcount(void), I make the following changes to the source files of xv6, in a top-down order.

In user.h, add int getreadcount(void); which is a C function declaration;

In syscall.h, add #define SYS\_getreadcount 22 which define a system constant for getreadcount syscall;

In syscall.c, add extern int sys\_getreadcount(void); which prepares for syscall by linking the C function int getreadcount(void) to the C function pointer;

In syscall.c, add [SYS\_getreadcount] sys\_getreadcount, within static int (\*syscalls[])(void) which adds the C function into the C function pointer;

In usys.S, add SYSCALL(getreadcount) which enables this newly added syscall to be invoked;

In sysfile.c, add a global variable static int readcount = 0;, add ++readcount; in the definition of int sys\_read(void) and define int sys\_getreadcount(void) by simply return the global variable readcount.

These changes enable us to add a new system call getreadcount, which works exactly the same as any other system calls such as read and write alike.

Both test cases passed.