Operating System

Report for Hw1

資工二 B00902008 陳明汎、B00902064 宋昊恩

1. Implementation:

We will state the changes that we have made for this project step by step.

We decided the corresponding names for each function in different files in the very beginning. Take function `OSAdd()` as an example. `OSAdd()` will send the exception named `SC_OSAdd` to `exception.cc` when it is called by users, and its implementation function is called `SysOSAdd()` in `ksyscall.h`.

Functions used to deal with subtraction(`OSSub()`), multiplication(`OSMul()`) as well as division(`OSDiv()`) are similar to additon(`OSAdd()`). Besides, function `Print()` will send the exception called `SC_Print` to `exception.cc`, and its implementation function is called `SysPrint()`.

First of all, we expanded the content in `code/test/Start.S`, implementated `OSAdd`, `OSSub`, `OSMul`, `OSMul` and Print just like the code for Add.

Second, we increased the definations and prototypes in `code/userprog/syscall.h`. We defined the value for each exception. Furthermore, we announced the prototype of each function.

Third, we appended the code in `code/userprog/ksyscall.h`. There are some different between `SysPrint()` and other functions.

On one hand, we implemented `SysPrint()` with two methods, one is `kernel->machine->ReadMem()` for reading virtual memory, and the other is `kernel->synchConsoleOut->PutChar()` for writing character to the console display and return immediately.

On the other hand, we implemented other functions just as it should be. Take example, int `SysOSAdd(int op1, int op2){ return op1 + op2; }`.

By the way, we implemented `PutChar()` and `PutNum()` with `kernel->synchConsoleOut->PutChar()` for generating the results should look like.

At the end, we changed the content in `code/userprog/exception.cc`. Take `SC_OSAdd` for example. We implemented it just as what `SC_Add` looks like.

However, there is a little bit difference between case `SC_Print()` with other functions, which is at we conveying 'char*' as the first parameter for `SysPrint()`. Moreover, we used self-defined functions `PutChar()` and `PutNum()` to achieve the required statement.

2. Some of our thought:

B00902064:

This homework is, at first sight, very difficult to implement. The first challenge that I encountered is the installation of `nachos`. I want to install `nachos` in my computer to prevent the unstable network linked to 217. However, even I spent a lot of time on fetching other on-line resources, it didn't work at all. After that, I spent even time more on understanding the functional logic for this kernel, since I found no introduction for this operating system.

The next challenge I met was that I can not compile my code correctly when I want to implement the function `SysPrint()` in the file `code/userprog/ksyscall.h`. After the assistance of my classmates, I finally found out the reason. The reason is that if I want to implement `SysPrint()` in that file, I must include `kernel.h`. However, the variable `ConsoleOutput` in `code/userprog/syscall.h` will conflict with class `ConsoleOutput` in `code/machine/console.cc`. After fixing this bug, I can print out the required statement correctly.

It is quite sad that I heard the TAs will introduce the usage of `nachos` after I had finished my homework...

B00902008:

It is really difficult to implement at my first glance because it is totally new platform that I had never contacted before. The first challenge that I encountered was to understand everything about what platform `nachos` do and how it work. Although I had read the spec carefully and exhaustively, I was totally confused at all until my teammate taught me in detail.

The next challenge I met was same as that my teammate met, couldn't compile changed code, `SysPrint()` in `code/userprog/ksyscall.h`, as we thought.

After receiving the helping hand from my classmate, we finally found out the reason. The reason why cannot compile as we thought is that it generated conflict between `code/userprog/syscall.h` with `code/machine/console.cc` when including `kernel.h` in `ksyscall.h`.

After fixing this bug, I can achieve the required statement correctly as the spec said.

3. Reference:

We get gigantic help from one of my classmates. We also find other detailed information from the Interet.