103011227 周延儒 資工 18

1. [20 points] 2.7: What are the two models of interprocess communication? What are the strengths and weaknesses of the two approaches?
   First of all, shared-memory communication is one of th communication. And, the pros is that it is faster than message passing model when the processes are on the same machine. On the contrary, the con is that different processes have to ensure they are not writing to the same location at the same time. Besides, processes also need to deal with the problem of memory protection and synchronization.

   Secondly, the message-passing model is also a kind of interprocess commination. The advantage is that it is easier to implement than shared-memory communication. The disadvantage, however, it is slower due to the time for connection setup.

2. [20 points] 2.10: What is the main advantage of the microkernel approach to system design? How do user programs and system services interact in a microkernel architecture? What are the disadvantages of using the microkernel approach?

   Well, in my opinion, the main advantage is that the new service does not require one to modify the kernel. It is much safer when more operations are done in user mode, compared with that in kernel model. In addition, the simpler kernel design can ensure more reliable OS.

   Moreover, the user programs and system services interact in microkernel architecture by using interprocess communication mechanism for example message.

   Last but not least, the disadvantage is the overhead that it is correlated to the interprocess communication and the constant use of the operating system’s message function to enable user process and the system service to interact with each other.

[5 points] On line {test}`/start.S:48`, you have the MIPS assembly instruction

```
addiu $2, $0, SC_Halt
```

which is another way of saying

register2 = 0 + SC_Halt;

Explain the purpose of this assignment statement.

Ans: It put system call number in register 2

The user Mips code that invokes syscall, first loads Mips register 2 with the integer that identifies the particular system call being invoked. Then it executes the syscall instruction.

2.4.1 Answer the following questions for the `Write()` system call. Hint: the answers can be found in the source code of exception.cc, including the comments!

- [5 points] How does the kernel get the pointer to the array of bytes to write? Give the description and C code.

  Ans:

  First of all, the argument in the function corresponds to the registers. To be more specific, the comments in the file address :

  ```
  //        arg1 -- r4
  //        arg2 -- r5
  //        arg3 -- r6
  //        arg4 -- r7
  ```

  As a result, to achieve the goal, we need to get the parameter of the Write() function, which is the arg1. And it is stored $4. Besides, the pointer is recorded in the memory of the Nachos, and we have to convert the string into the real address in linux.

  ```
  val = kernel->machine->ReadRegister(4); //arg1 "hello world"
  char *msg = &(kernel->machine->mainMemory[val]);//pointer
  ```

- [5 points] How does the kernel get the number of bytes to write? Give the description and C code.

  Ans:

  Similary,we can geth the number of bytes to write by get the value of arg2. And it is stored in $5

  ```
  int size=kernel->machine->ReadRegister(5); //arg2 12
  ```

```
DEBUG(dbgSys, "number of bytes: "<<size);
```

- [5 points] How does the kernel return the value to the caller? (for the number of bytes written?) Give the description and C code.

  Ans:

  Kernel return the value to the caller by writing the value to the &2. That is to say, the system call.

```
status = kernel->machine->ReadRegister(5);
kernel->machine->WriteRegister(2, (int) status);//cp size and return
```