103011227 周延儒

[20 points] 3.2: Describe the actions taken by a kernel to context-switch between processes

First of all, the OS must save the state of the running process currently and then restore the state of the process schedules to be proceed next.

Besides, saving the state of the running process usually includes the values of all CPU registers and memory allocation as well.

Still, context-switch has to perform many architecture-specific operations such as flushing data and instruction caches

[20 points] 3.11.a: What are the benefits and the disadvantages of each of the following? Consider both the system level and the programmer level.
a. Synchronous and asynchronous communication

Well, the advantage of the Synchronous communication is that it offers rendezvous between the both sender and the receiver. The disadvantage of a blocking send, on the other hand, is that the rendezvous might not be required and the thus the message could be sent asynchronously. Thus, the message-passing systems usually offer both forms of synchronization.

[20 points] Explain why dereferencing the first parameter in ExceptionHandler() does not result in the string whose address is passed by the statement Write("Hello world\n", 12, 1);
in {test}/hw2.c:main() .

Well, the key to the problem is actually mentioned in the previous assignment. To be more specific, this is because the pointer value to the string, the first parameter, is actually the address value under the Nachos mainMemory instead of the real Linux address value. As a result, dereferencing the value will cause core dump.

[20 points] Explain the correct way for ExceptionHandler() to obtain the string, by giving both verbal description and the C code to do it

To solve the problem is quite straightforward. All we need to do is convert the value to the real value in the base Linux address value .

```
val = kernel->machine->ReadRegister(4); //arg1 "hello world"
char *msg = &(kernel->machine->mainMemory[val]);//pointer
//hw3 the entire string
DEBUG(dbgSys, "The entire string : " << msg);
```