CIS 415 – Operating Systems

Homework Assignment 1
Fall 2016 – Prof. Sventek

Textbook Questions (30 points)

1. OSC 2.18: What are the two models of inter-process communication? What are the strengths and weaknesses of the two approaches? (6 points)

   The two models of interprocess communication are message-passing model and the shared-memory model. Message passing is useful for exchanging smaller amounts of data, because no conflicts need be avoided. It is also easier to implement than is shared memory for intercomputer communication. Shared memory allows maximum speed and convenience of communication, since it can be done at memory transfer speeds when it takes place within a computer. However, this method compromises on protection and synchronization between the processes sharing memory.
2. OSC 2.19: Why is the separation of mechanism and policy desirable? (4 points)

   Mechanism and policy must be separate to ensure that systems are easy to modify. No two system installations are the same, so each installation may want to tune the operating system to suit its needs. With mechanism and policy separate, the policy may be changed at will while the mechanism stays unchanged. This arrangement provides a more flexible system.
3. OSC 3.9: Describe the actions taken by a kernel to context-switch between processes? (4 points)

   In general, the operating system must save the state of the currently running process and restore the state of the process scheduled to be run next. Saving the state of a process typically includes the values of all the CPU registers in addition to memory allocation. Context switches must also perform many architecture-specific operations, including flushing data and instruction caches.
4. OSC 3.18: What are the benefits and the disadvantages of each of the following? Consider both the system level and the programmer level. (16 points)
   a. Synchronous and asynchronous communication - A benefit of synchronous communication is that it allows a rendezvous between the sender and receiver. A disadvantage of a blocking send is that a rendezvous may not be required and the message could be delivered asynchronously. As a result, message-passing systems often provide both forms of synchronization.
   b. Automatic and explicit buffering - Automatic buffering provides a queue with indefinite length, thus ensuring the sender will never have to block while waiting to copy a message. There are no specifications on how automatic buffering will be provided; one scheme may reserve sufficiently large memory where much of the memory is wasted. Explicit buffering specifies how large the buffer is. In this situation, the sender may be blocked while waiting for available space in the queue. However, it is less likely that memory will be wasted with explicit buffering.

c. Send by copy and send by reference - Send by copy does not allow the receiver to alter the state of the parameter; send by reference does allow it. A benefit of send by reference is that it allows the programmer to write a distributed version of a centralized application. Java's RMI provides both; however, passing a parameter by reference requires declaring the parameter as a remote object as well.

d. Fixed-sizes and variable-sized messages - The implications of this are mostly related to buffering issues; with fixed-size messages, a buffer with a specific size can hold a known number of messages. The number of variable-sized messages that can be held by such a buffer is unknown. Consider how Windows 2000 handles this situation: with fixed-sized messages (anything < 256 bytes), the messages are copied from the address space of the sender to the address space of the receiving process. Larger messages (i.e. variable-sized messages) use shared memory to pass the message.

Process Analysis (20 points)

Suppose you want to find out as much as you can about a running process, one where you do not have access to the source code. Describe a procedure that uses files in the Linux proc filesystem[1] to extract as much information about the process as you can. You should be able to apply this to any of your currently executing processes. For instance, an experiment may begin something like this:

```
% sleep 600&
[1] 1363
. . .
```

You can cd to /proc/1363 and run ls, where you will see a number of files and directories. Some of the files contain relevant information. Explain which piece of information each step of your procedure is designed to supply, and how it relates to the program, process, and operating system.

Expected to see discussion of /proc/PID/cmdline, /proc/PID/exe, /proc/PID/status, /proc/PID/stat, /proc/PID/io, and /proc/PID/maps. status and io will be most useful for process scheduling.

Note: Like all assignments in this class **you are prohibited from copying any content from the Internet or sharing ideas, code, configuration, text or anything else or getting help from anyone in or outside of the class, except where noted**. Consulting online sources is acceptable, but under no circumstances should anything be copied. Failure to abide by this requirement will result in sanctions ranging from zero on the assignment to dismissal from the class.

[1] See http://man7.org/linux/man-pages/man5/proc.5.html