

Chapter 2 Reading Questions:

2.13) Describe the three general methods for passing parameters to the operating system

Response:

- Programs can push parameters onto the stack, while the OS can pop parameters off the stack
- Parameters can be passed inside of registers
- Registers can also store the addresses of blocks of parameters

2.14) Describe how you could obtain a statistical profile of the amount of time a program spends executing different sections of its code. Discuss the importance of such a statistical profile.

Response: The intended value of a statistical profile of time spent executing code is the same as any statistical profile of an executing program: optimization. With the time profile, the programmer would be able to gauge the actual runtimes of specific code to determine if snippets of code should be given more thought for optimization. Generation of the time profile can be done through a tracing facility or regular timer interrupts that would record the value of the program counter. The time required by a section of code could also be measured through use of the `time()` system call before and after code execution and computing the difference.

2.18) What are the two models of interprocess communication? What are the strengths and weaknesses of the two approaches?

Response: The two common models of interprocess communication are the message-passing model and the shared-memory model. Message passing (as capitalized on in Erlang) is highly efficient in passing smaller amounts of data, though is slower at passing data than shared-memory and requires some connection to be established before communication can begin. Shared-memory allows for maximum speed, however much care needs to be taken in protecting and synchronizing the processes shared-memory accesses.

2.21) What is the main advantage of the microkernel approach to systems design? How do user programs and system services interact in a microkernel architecture? What are the disadvantages of using the microkernel approach?

Response: The main advantage of the microkernel approach to systems design is that expanding the operating system in the future is much easier than alternatives; all new services are added to the user space and do not require modification of the kernel itself. User programs and system services communicate through message-passing; which is the main function of the kernel, coordinating the message-passing. As most of the services are running as user processes and only messages are passed to the kernel, microkernel systems tend to be more secure and reliable. The main disadvantage of the microkernel approach is that the message-passing system mechanisms are used constantly for communication between user processes and system's services.

2.16) What is the relationship between a guest operating system and a host operating system in a system like VMware? What factors need to be considered in choosing the host operating system? [In the 9th edition of the textbook, VMWare is discussed in Chapter 1 (pg. 41-45).]

Response: Like all good hosts for their guests, host operating system says "come make yourself at home" to the guest operating system and offers up all of the services and functionality that the host normally

controls. The guest moves in and takes all of the services that it normally would provide and maps it on to services provided by the host. The biggest factor to consider when choosing a host operating system is the amount of functionality that the guest operating system can interface with; with a greater amount of interfaces available, a more diverse amount of guest operating systems can be mapped onto the host operating system.