# Comprehensive Exam: Software Systems Solutions (60 points)

1) **(10 points) Describe the purpose of device drivers in modern operating systems. (Hint: I'm looking for something more than "they drive devices.")**

The key purpose of device drivers is to keep device-specific algorithms and data structures contained to a relatively small portion of the kernel. The operating system defines a standard interface to devices and it is the job of the device drive to convert this standard interface to the needed device-specific operations.

2) **(18 points) The algorithms and data structures used in operating systems are frequently a function of the hardware technology. As hardware technology has changed so have operating systems. For the following hardware technology changes, describe the operating system algorithms and data structures that might need to be changed.**
**(a) A very large (multiple order-of-magnitude) increase in physical memory size.**
**(b) A large increase in the number of bits in a virtual address.**
**(c) An increase in the number of CPUs in the system from one to many tens of CPUs.**

(a) A large increase in physical memory will stress the physical memory management data structures including the memory free list and the page replacement algorithm. Since memory is a less precious resource, we might want to modify the OS to spend less resources tracking it. For example, switching to a simple page replacement algorithm rather than LRU. Other algorithms that would probably need to be changed including the management of swap space (physical memory might be larger than the swap disks). Additional caching of files, etc. could be done with the extra memory.
(b) A large increase in virtual address space will stress the virtual to physical mapping data structures (i.e. page tables) maintained by the kernel. The OS will have to go to a multi-level or invented page tables. It can also cause problems for swap space management.
(c) Large changes will be needed in the synchronization used as well as the scheduling and many other parts of the system.

3) **(12 points) The UNIX operating system has a file buffer cache that uses a writeback policy with a 30 second timeout. This means that changes to file blocks can sit in the buffer cache for up to 30 seconds before being written back to the disk. What are motivations for and drawbacks of this scheme?**

The motivations include decoupling of file write speed from that of the disk, the ability to coalesce multiple writes to the same block into a single write to disk, and the ability to generate multiple block writes so the disk scheduler can optimize disk access patterns. The chief drawback is data can be lost if the system crashes.

4) **(8 points) Describe what an atomic operation is. Give examples and describe why they are useful.**

An atomic operation is an operation that, from some point of view, appears indivisible. Partial results of an atomic operation are never visible. From another entity viewing the operation, the changes made will happen instantly so the viewer will see either all or none of the changes. Examples of atomic operations are operations on a data structure such that all operations are performed while holding a lock. Atomic operations make it easier to reason and build systems because within atomic operations the presents of other processes can be ignored.

68

5) (12 points) Describe the necessary and sufficient conditions for deadlock. For each condition describe a practical deadlock prevention technique that works by preventing the condition from arising.

There are four necessary and sufficient conditions for deadlock:
(a) Limited access: resources cannot be shared.
(b) No preemption. Once given, a resource cannot be taken away.
(c) Multiple independent requests: processes don't ask for resources all at once.
(d) There is a circularity in the graph of who has what and who wants what.

Preventing the condition:

Violating (a) is pretty tough to do in a general-purpose system. Having enough resources so that they don't need to be shared is done is some special-purpose operating systems.

Violating (b) is again pretty difficult in a general-purpose system. Although it is possible to construct a system such that all resources are preemptable, resources attached to the outside world (e.g. terminals, printers) are difficult to preempt.

(c) can be violated by requiring processes to request all resources at once and having the system wait until all the resources are available before continuing.

(d) can be violated by forcing an order to resource requests such that cycles in the graph are avoided.

69