

EEE6032 - Operating Systems - Model Answers

Q1:

(a) Any three of enabling/disabling interrupts, performing an I/O operation, executing halt/stop command.

All these instructions share the property that they affect non-shareable resources of the computer. [5 marks]

(b) A user process can access a privileged instruction via a system call, typically invoked with a trap command, execution of which places the processor in system mode. The parameter of the trap instruction indicates to the kernel that a system call is being requested, and this invokes an interrupt service routine (ISR) which determines the exact system call being requested by examining the contents of a specific register loaded with a ID before the trap is executed.

This ID typically is the index into a LUT of system call entry points. The appropriate system call, ^{containing the privileged instruction(s),} is then executed.

To pass parameters to a system call, the user program

would load pre-determined registers with the necessary values. The ISR would then push these values onto the kernel stack before passing control to the system call which would access the parameters on the kernel stack.

Eventually, the system call would return to the ISR which would return to the user process, simultaneously putting the processor in user mode. [6 marks]

(c) In order to execute an I/O operation, the user process would have to execute the appropriate system call. This system call would perform the I/O but instead of returning to the user process (which might have to idle, waiting on the I/O), the I/O system call would invoke the OS's process scheduler which would perform a context switch. [5 marks]

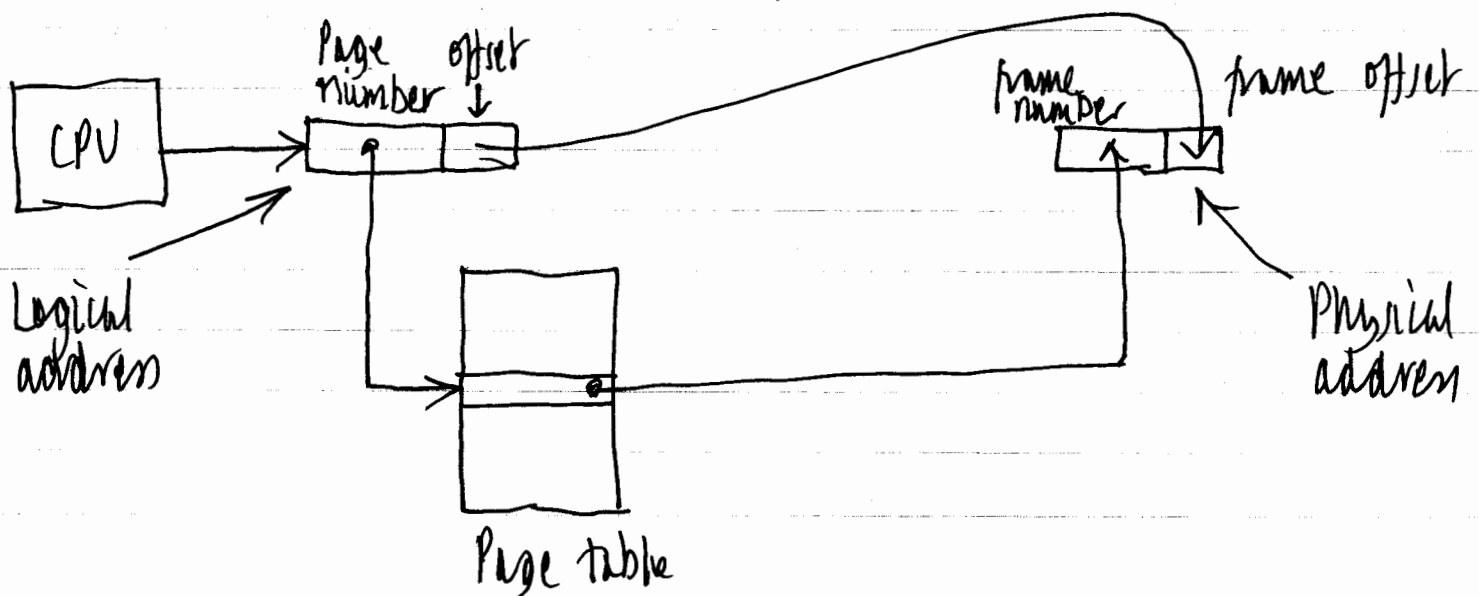
(d) Generally, ^{file} ~~an~~ I/O would not invoke a context switch since modern file systems buffer files in memory. Thus, in general, only an input operation for which the data was not cached.

would result in a context switch. Most other file operations would require only memory reads from the cache. [4 marks]

Q2:

(a) A logical address space is the space of addresses generated by the CPU. A physical address, on the other hand, is the address in actual ~~RAM~~ RAM. A logical address and a physical address are usually different but maintain a unique, 1-to-1 correspondence. [2 marks]

(b) In order to perform the translation between a logical and a physical address, the logical address would be partitioned into a page number and page offset.



(4)

The least significant bits would constitute an offset within a page and be copied directly to the physical address. The most significant bits of the logical address form the index into a page table which yields the frame number and the most significant bits of the physical address.

This translation process allows the programmer to see a contiguous logical address space which maps to pages scattered (anywhere convenient) in ^{physical} memory. [8 marks]

The problem with using very small pages is that every page needs one entry in the page table. Small pages therefore result in a very large amount of memory being required for the page table.

Multi-level page tables ^(typically) reduce the amount of memory required ~~for the page~~, because not all levels in the page table will be populated and hence require the allocation of physical memory.

In principle, multiple level page tables increase the effective addressing time since one memory access is required ~~for~~ to retrieve the table entry at every level. In practice, the Principle of Locality together with caching (e.g. Translation Lookaside Buffers) mean this disadvantage does not have too great an impact.
[4 marks]

(c) Under the classical UNIX fork, ~~the~~ an exact copy of the parent process is made to comprise the child. This involves a byte-for-byte copy of the parent process. This copy is time-consuming and typically wasted if the child immediately overwrites itself with the image of a new process (as is commonly the case). In a COW arrangement, the parent is not copied - parent and child share the same memory - ~~but~~ but that memory is tagged as 'copy-on-write'. If either child or parent attempts to write to the memory, only then is a copy made to avoid a conflicted version.

⑥

Since a write operation would require substantial processing before it could be executed, a fault would be raised. On return from the ISR, the write operation would be retried. [6 marks]

Q3:

(a) It would be possible to run the executable code on a virtual machine which emulated the instruction set^(and system calls) of the obsolete computer. Each instruction from the obsolete computer would have to be translated to the equivalent (or equivalent set of) instructions of the host hardware. As a consequence the runtime performance would be degraded compared to running native code. [4 marks]

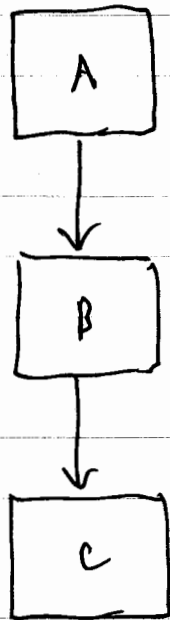
(b) Under windows, a child process would be created with the `CreateProcess()` API function. This takes the filename of the child process executable as its principal parameter. Other parameters would specify: the execution model, the inheritance of resources, [4 marks]

and security attributes.

(c) Certain high-priority kernel processes are scheduled as first-come-first-served because their execution is critical to the efficient running of the OS. FCFS assures rapid throughput without wasting unnecessary CPU time on context switches.

[4 marks]

(d) Diagrammatically;



If B terminates, the OS will kill C since its continued existence is pointless. A would be notified of B's termination, possibly by a signal.

[4 marks]

(e) The advantage of shared memory as an IPC mechanism is its simplicity. The shared memory block is accessible as a linear array. Compared to pipes, ~~that~~ which are strictly sequential in nature, shared memory can be accessed randomly.

[4 marks]

⑧

Q4:

(a) Cross compilation is the building of code for a target system on a ~~base~~ host system with a different architecture and/or operating system. Cross compilation can be useful for:

i) Embedded development where comprehensive programming tools are not available for the target machine.

ii) Producing multiple versions of an executable for different platforms: the build process can be configured to generate all versions in one go. [3 marks]

A build system is the system on which the cross-compiler is compiled/built.

A host system is the system on which the cross-compiler is run.

The target system is the system ~~to~~ on which the developed software will be run.

[3 marks]

(9)

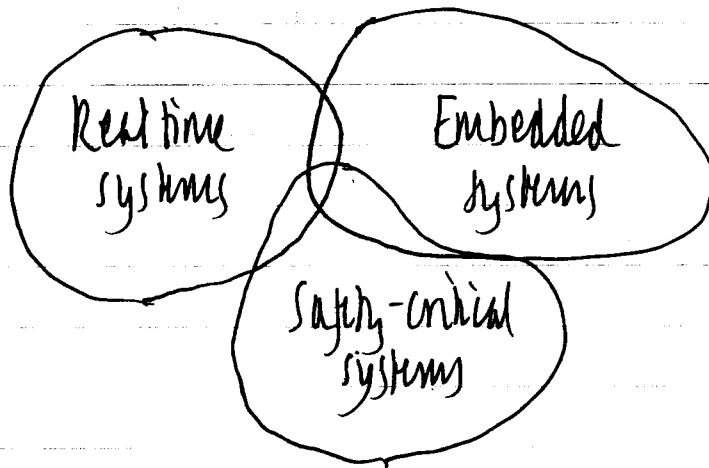
(b) The advantage of using shared object libraries is that if several processes access its functions, a single copy can be loaded into memory, thereby saving memory. In contrast, in static linking each process would load identical copies of the functions. In practice, the above advantage is only realised if ≥ 2 processes share functions.

A shared library incurs two runtime penalties: First, if the SO library is not loaded, an appropriate OS process must be invoked to load it. Second, the entry point to a function is determined by two levels of indirection leading to a minor increase in runtime. [4 marks]

Under UNIX, a shared object library is automatically loaded, on demand, by the OS. A dynamic link library is explicitly loaded by the program. DLLs can be useful to run rarely accessed functions, such as unusual error conditions, but not loading them ~~too~~ permanently into memory. [4 marks]

10

(c) The differences between these frequently-confused terms can best be illustrated by:



Real-time systems are required to meet a deadline. Embedded systems typically control some other system. Safety-critical systems operate in an environment where failure is disastrous.

Although these classes often overlap, they are distinct. [2 marks]

Interrupt latency is the delay between receipt of an interrupt and the machine starting to run the associated ISR. The dispatch latency is the delay between pre-empting one process and the commencement of the next.

Both types of latency can be minimised by using a fully pre-emptible kernel.

[4 marks]