

CS 385: HW1–Due Feb 24, 2004

Jon A. Solworth

February 16, 2004

1. Real, virtual, transparent. For the following, describe which of the adjectives of real, virtual or transparent applies:
 - (a) You just replaced your disk drive with a flash memory device which emulates a disk drive but is much faster.
 - (b) You just bought a new disk controller, which has a large memory on it enabling it to perform read operations without accessing the disk. Your software is unchanged.
 - (c) You just signed up for a web hosting service. The web service appears under your own unique domain name but actually one computer supports multiple domain names.
2. Finite progress. Do the following implementations provide finite progress or not. Explain.
 - (a) A timer interrupt occurs every $1/100$ th of a second unless a `postponeInterrupt` system call. When the `postponeInterrupt` is invoked, it extends the time until the next timer interrupt by another $1/100$ th of a second and prevents a context switch.
 - (b) Process i gets $\frac{1}{N}$ th the CPU cycles, when there are N processes.
 - (c) Only when a process does not execute at least one system call per millisecond, is it not context switched.
3. Privilege mode.
 - (a) In what software is privilege mode on or off.
 - (b) Describe all the ways that privilege mode is turned on.
 - (c) Describe all the ways that privilege mode is turned off.
 - (d) How can privilege instructions be created and where can privilege instructions be used.

4. Describe what happens from the time the computer is turned on through the time the init process begins executing. Be very detailed with describing what happens at the processor, the bios, the disk, and with the init process.
5. How does your computer know how to load different OS (eg. Linux or Windows) into the computer's main memory? How do you load the OS without their being an OS present in main memory?
6. Why does a trap instruction specify a trap number rather than a trap address? What happens if you just do an absolute branch back to user code rather than a rtt? Explain.
7. Take the syscall implementation on page 21, and add in code necessary to work with base and limits register. Does the RTI need to modify base and limits register as well? Show how it would be modified.
8. When and how can a context switch occur? Can a process prevent a context switch from occurring? Explain.
9. What happens on a system call which is not specific to the system call being processed.