CSE 121: Homework 3

Due: Thursday, 3/5/2009, 11am pst (email kaisenl@cs)

Note 1: If emailing, use PDF, not DOC/DOCX (Google for CutePDF if you need a PDF printer)
Note 2: Answers quoting passages (verbatim or close to verbatim) receive no credit. Summarize.
Note 3: Out of 60 pts. You may skip 3 problems. Please designate which problems you are skipping. Keep in mind that although you can skip a paper entirely, you are still responsible for knowing it on the final.

1. **Lottery Scheduling**

   (a) Lottery Scheduling is a proportional-share scheduler meaning amount of CPU time received over time is proportional to the shares allocated. How does the use of compensation tickets help achieve this?

   (b) It is important to minimize the work done during schedule operations because it is pure overhead. One optimization is to put tasks with more tickets in the front of the list when scanning through, and another is to "move to front". Why do these techniques speed up lottery winner selection?

   (c) Suppose you're working in Eclipse and watching a movie at the same time. The Eclipse application forks off multiple processes like the static analyzer and incremental compiler. Similarly, your media player forks off a DVD decoder and audio decoder. You want to give both applications the power to create tickets and schedule their own processes. However, you are worried that the media player might create more tickets and you won't be able to work anymore (the horror!). What can you do with the lottery scheduler to prevent this?

2. **Mach**

   (a) Part of Mach's success was attributed to its ability to run BSD binaries. Explain how this is done.

   (b) Mallory's new Mach-compatible malware application has accidentally made it onto your machine. It attempts to send a continuous stream of spam by flooding your network manager with IPC messages. How does Mach protect against this?

   (c) Suppose there is a user-level task that runs after bootup called INIT. INIT is responsible for starting the file system, window manager, mail client, etc. Assuming INIT does not know all the port relationships a priori, how would you allow the file system to communicate with the disk driver and the window manager to communicate with the screen?

3. **L3/L4**

   (a) Lietdke believes that only the bare minimum set of abstractions should be in the microkernel. What are the two abstractions and why did he decide this?

(b) L3/L4 achieves high performance by sacrificing portability to different processor architectures. Ben Bitdiddle wants to use L3/L4 on his iPhone and is being paid to optimize performance as much as possible. The iPhone has an ARM1176JZF processor: 32-bit address space, fully associative MicroTLB with an Address Space IDentifer (ASID) for each entry, 4-way set associative cache with virtually indexed physically tagged entries, tightly coupled memory (low latency memory without cache unpredictabiliity), and multi-level page tables. Give 3 suggestions to Ben as to how he can optimize his L3/L4 implementation. [Hint: There are many answers, not all may be in the paper.]

(c) What does figure 4 tell you about microkernel construction regarding cache use? [Hint: Think back to lecture]

4. **Nooks**

(a) Nooks is designed to not only detect device driver errors, but also recover from them (or at least prevent the system from crashing). Briefly describe the phases that recovery goes through and why each phase is necessary.

(b) It seems like the authors could have increased protection by moving extensions outside the kernel all together (similar to what they did in Mach/L4). Instead, Nooks uses a lightweight kernel protection domain to isolate extensions. What are the benefits of keeping extensions in the kernel?

(c) Figure 6 shows the reliability of various extensions with and without Nooks. Why was it much easier to crash the network driver compared to the file system? Suppose you attach a GPS to your computer. If you ran the same evaluation as in Figure 6, do you think the effectiveness of Nooks would be big (like in pcnet32) or small (like in VFAT)? Justify your answer.

5. **Exokernel**

(a) The Exokernel aims to bring the abstraction level all the way down to multiplexing the physical resources themselves. What would be the Exokernel-equivalent abstraction for common abstractions such as a file system, TCP streams, address spaces, and a window manager?

(b) Secure bindings aim to separate authorization from resource use. The paper gives two example uses of secure bindings: the TLB and packet filter. Assuming you had a hardware device that only had a file system abstraction (as opposed to a disk block abstraction), what secure binding mechanism could you use for a file system abstraction?

(c) Ben Bitdiddle wants to use ASHs for his packet filter. He wants to download application code into the kernel, but is worried about security. He requires that all handlers be written in Java. Should he still be worried? Justify your answer.