Mittuniversitetet
MID SWEDEN UNIVERSITY

Final exam

DT011G Introduction to Operating Systems

Daniel Bosk
daniel.bosk@miun.se
Phone: 060 - 14 8709

2013-01-18

Instructions

Carefully read the questions before you start answering them. Note the time limit of the exam and plan your answers accordingly. Only answer the question, do not write about subjects remotely related to the question.

Write your answers on separate sheets, not on the exam paper. Only write on one side of the sheets. Start each question on a new sheet.

Make sure you write your answers clearly, if I cannot read an answer the answer will be awarded no points – even if the answer is correct. The questions are *not* sorted by difficulty.

Time 6 hours.

Aids Dictionary, course literature [SGG09], graded assignments and personal notes.

Maximum points 25

Questions 5

Preliminary grades

The following grading criteria applies: E $\geq$ 50%, D $\geq$ 60%, C $\geq$ 70%, B $\geq$ 80%, A $\geq$ 90%; with no question awarded zero points.

Aim

The aim of the exam is to examine that you have fulfilled the requirements specified in the course syllabus.

An operating systems oriented computer science fiction, or: The exam questions

(4p) 1. At your work there is this guy Bill, he always comes to you for help. He is a nice guy, he always means well. But sometimes things just does not turn out as intended, alright, most of the times. This time, however, he comes with a minor problem, it is not a problem really, more of a thought.

"Sorry to bother you again", he says, "I was looking around for ideas on how to improve the built-in web browser in my system", Bill maintains an operating system in his spare time, "and I was looking at Firefox and Chromium. As I understand it, Firefox runs each tab as a new thread whereas Chromium spawns a new process for each tab. Which approach do you think would be the most advantageous?"

Help Bill by explaining the advantages and disadvantages of each approach.

(5p) 2. By the end of the day your telephone rings, it is your not-so-tech-savy friend. He is by the way a rather enthusiastic fellow, some would even go for the word excentric. His name is Steve, and you are going out with Steve and Bill to catch the latest film in the theatre right after work.

"How're you doing", he says, "are you and Bill on your way?"

"Yes, we'll leave in a few minutes", you say.

"Great! You know I was thinking about what you said about deadlocks a few months ago. I'll tell you when you get here."

You and Bill leave the office and arrive at the theatre to meet Steve by the registers to get your tickets.

"About that deadlock-thingy, I was thinking about getting these tickets. You know, the cashier does not trust us to give us the tickets before we pay. Hence, we have to trust the cashier by paying and then hoping we get the tickets. If we would not trust the cashier, and thus would require the tickets before we pay, *then* we would encounter a deadlock, isn't that right?"

It would appear Steve actually learned what a deadlock is. But just to be sure he is not just lucky, talk him through why this is a deadlock.

3. After the film has ended you pose the question about threads and processes to Steve.

"Steve, Bill is thinking about whether to choose threads or processes in his browser, what are your thoughts?"

"Certainly he should choose the one easiest for developers. Getting developers to your platform is the key to success", Steve replies.

"Sure, but shouldn't I think first about my users", Bill asks.

"Developers, developers, developers", Steve starts to chant like obsessed, shortly continuing to shouting, "DEVELOPERS, DEVELOPERS, DEVELOPERS! I love that word; some day, if I am to hold a presentation I will start it with those very words."

"We know that for sure, and so does the rest of the theatre. I solved the networking stuff by copying the the code from Berkeley, but I do not know how to solve the interface to the hardware. I have heard of monolithic kernels, loadable kernel modules and drivers; I do not really know what that means though."

You on the other hand, know perfectly well the different approaches to interface the hardware. Explain to Bill

(2p) (a) the different types of kernels there are, and

(4p) (b) what advantages there are of each type.

4. You are at work again. Bill is busy working, in the evenings he is implementing the things you have talked about earlier; this is very nice as this allows you to actually get some work done. However, you are very good at distracting yourself on your own as well. You hear a colleague of yours moaning aloud about how "it is time to defrag again".

Explain to your colleague

(2p) (a) what defragmentation is and why it is necessary on some file systems,

(2p) (b) tell him about some of the file systems which does not require defragmentation (this includes why they do not need this).

(6p) 5. You are back in your office, it is Friday afternoon. Another one of your co-workers comes by. He says to you "I hope you haven't forgotten about our meeting. You promised me you'd give me some feedback on my ideas for Freax", and he smiles. He knows just as well as you do that you did forget about it, it was not in your calendar and you have been so distracted this week you remember only fragments.

"Well," he continues, "we are supposed to discuss some aspects of scheduling for my free-time project." His name is Linus and he is an intern from the University of Helsinki. "What I'd like to do is to have a nice general scheduling algorithm for processes. Interactive processes should have priority over I/O-bound processes". The two of you enter the meeting room.

Give Linus an overview of how to implement his scheduler. (A tip is to look at how the Linux operating system actually does its process scheduling; yes, there is a little "Back to the future" about this tip.)

Finally Friday evening, now it is time for a treat after all this hard work.

The end.

References

[SGG09] Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne. *Operating System Concepts*. John Wiley & Sons Inc, Hoboken, N.J., 8 edition, 2009. International Student Version.