Term Project

Proposals due: 11:59 PM, Thursday, April 14, 2005
Progress reports due: 9:55 AM, Tuesday, April 26, 2005
Paper draft due: 11:59 PM, Tuesday, May 3, 2005
Presentations: Tuesday, May 10, and Thursday, May 12, 2005
Final paper and software submission: 2:30 PM, Friday, May 13, 2005

You may choose your own topic for the final project. Choose a topic related to operating systems and study it in more detail than we have in class. Most projects will include significant programming, but the amount may vary from topic to topic. A formal research paper and brief in-class presentation are required.

You may work individually or in groups of two or three for the term project. Groups must be formed by 9:55 AM, Tuesday, April 12, 2005.

The proposal

Please start thinking about a topic right away and come to me with your ideas. By 11:59 PM, Thursday, April 14, 2005, submit a proposal, no more than one page in length, that describes your topic, what you plan to investigate, and how you plan to go about it. If you will need access to any special hardware or software, include that in your proposal. Submit as a PDF file `proposal.pdf`.

The progress report

By 9:55 AM, Tuesday, April 26, 2005, submit a progress report. This should consist of an early draft of your paper including sources, and a description of the design and current implementation status of your software, as well as a timetable for completion of the project. Submit a PDF file `progress.pdf`.

The paper

This is to be a formal research paper, and should be organized as such. You should begin with a title, author list, and abstract. The main body of the paper should be organized into sections including (*i*) an introduction in which you describe the general topic and the particular aspects you will be examining, (*ii*) one or more sections comprising your main text, where you describe what you have done, how you have done it, and what you have learned, (*iii*) a conclusion, which should include ideas for future investigation into your topic which were beyond the scope of your paper, and (*iv*) a complete list of citations. Citations of web pages are acceptable in some circumstances, but books, articles in conference proceedings or journals, or technical reports are preferred. Examples of papers that fit the expected format will be provided.

Proper English and a good technical writing style are important. Writing well is very difficult – it is an iterative process and cannot be done all at once. Be precise and be concise. Group members should proofread and make suggestions about each other's writing. Check your spelling and grammar carefully. I expect most papers will be around 15 single-sided pages, using 1.5 spacing, one inch margins, and a 12-point Times Roman font (or as close as you can come). Please do not adjust margins and font sizes to force a certain length. You are encouraged but not required to use LaTeX to typeset your paper. The LaTeX example in my shared area is already set up with an appropriate format. Length is not important – content and quality are. Papers shorter than 10 pages or longer than 20 pages are acceptable, if the length is appropriate for the content.

Submit a draft, as complete as possible, as a PDF file `draft.pdf` by 11:59 PM, Tuesday, May 3, 2005. The more complete this draft, the better feedback I can give you to improve it. You may also submit additional drafts for feedback, but keep in mind that it may take me a day or two to get to them. Submit your final version as a PDF file `paper.pdf` by 2:30 PM, Friday, May 13, 2005.

The project

You should submit your source code and instructions on how to build and run the software that you develop or modify for this project. The software should be described in the paper. You should make arrangements to demonstrate when you have done. Source code should be submitted and demonstrations completed by 2:30 PM, Friday, May 13, 2005.

The presentation

Each group will present a summary of their work to the class. Include background information on your topic, a description of what you did, and a summary of what you learned. Software demonstrations may also be appropriate. Prepare slides or web pages. Rehearse your presentation, paying special attention to timing. Our schedule will be tight, so groups will not be allowed to run over the alloted time. Depending on the number of groups, there may be as little as 15 minutes available per group for your presentations. Presentations will take place in class on Tuesday, May 10, and Thursday, May 12, 2005, with time slots chosen in class the week before. All group members must participate in the presentation. Attendance is required at both presentation sessions, not just when your group is speaking.

Grading

This final project accounts for 25% of the course grade. The grade will be based on all aspects of the project: the proposal (5%), the progress report (5%), the design (5%), documentation, style, and correctness of the software developed (up to 50%), the content and writing style of the draft (5%) and the final paper (up to 75%), and quality of the presentation (10%). Note that 75% of this project will be allocated between the software developed and the final report, with the exact ratio determined by the size and complexity of the programming task undertaken. No credit is given for attendance, but a penalty of 5 points will be applied for missing class on a presentation day.

Choosing a topic

Here are some ideas to consider when choosing your topic:

- Experimental modifications to FreeBSD or Linux kernel (such as custom schedulers, memory management, file systems, device drivers).
- A more detailed simulation (along the lines of the round-robin simulator) of some

aspect or aspects of operating systems.
- OS issues arising in high-end web servers or server farms (think Google, very popular news/sports/weather sites).
- Implementations of an applications of semaphores or other synchronization techniques.
- Experiments with how (insert OS here) handles (insert important OS issues here), possibly trying other possibilities.
- Mass storage systems, high performance file systems, file system safety and repair.
- OS issues in autonomous systems, embedded systems, or real time systems.
- Emulators or virtual machines – build your own or enhance an existing one.
- Protection and security issues, encryption.
- Distributed operating systems.
- File sharing over wide-area networks.
- OS issues in Grid computing, internet computing.
- Theoretical issues in OS, such as queueing theory.

These are just ideas, and you should not restrict yourself to this list. I hope everyone can find an appropriate topic that they find interesting.

A dedicated lab computer can be made available to each of a few groups who wish to pursue a project that requires root access. If you think you may be interested in this, be sure to stake your claim right away.

Honor Code Guidelines

Collaboration within a group is unrestricted. Since each group is working on a different project, you are free to discuss your projects with each other. If you wish to use or refer to any software libraries or outside source code beyond the standard language (C, C++, Java) libraries, check with me first. All sources must be cited properly. If in doubt about anything related to the Honor Code, ask now and avoid problems later!

Final Thoughts

You have six weeks, so the expectation is for six weeks of work. You will not be able to do a good job if you put it off. Don't end up having a conversation like this one when May comes around and the pressure is on:

Dr. Raymond Stantz: I tried to think of the most harmless thing. Something I loved from my childhood. Something that could never ever possibly destroy us. Mr. Sta-Puft!

Dr. Peter Venkman: Nice thinkin', Ray.

Dr. Peter Venkman: Ray has gone bye-bye, Egon... what've you got left?

Dr. Egon Spengler: Sorry, Venkman, I'm terrified beyond the capacity for rational thought.

– *Ghostbusters*