Project Guidelines and Schedule

You have the choice of either writing a survey paper or doing an implementation project. Survey papers are written individually whereas implementation projects should usually be done with a partner in groups of 2.

You can use the piazza forum if you have a project idea and are looking for a partner to share it with.

To ensure progress towards the completion of your project or paper, your team will meet with the instructor during certain milestones, listed below. Those meetings are mandatory for full credit on the project.

Timeline

All deadlines are 11:59pm on the day listed.

Oct 5	Initial proposal (max 2 pages) due. For survey papers, submit a description that describes your initial research on the topic, including a list of literature you will study. For a survey paper, I insist that you use a word processing system that separates the structure of your text from the formatting, such as LaTeX or Microsoft Word Templates. For projects, describe the goal of your project (what are you going to show?), the relevance (why is this important?), the major components and the technology infrastructure you plan on using. Include a plan of work. Outline what you will have accomplished by milestones 1 and 2 (see below.) You should have started initial investigations by that time.
Oct 8	Work with instructor to revise and get approval for topic.
Oct 26	*Survey:* Outline of paper due. In addition, create and submit written summaries of the sources you are researching. *Project:* Progress report 1. (1 page)
Oct 29	Meet formally with instructor to discuss progress.
Nov 9	*Survey:* First complete draft of paper due. *Project*: Progress report 2. (1 page)
Nov 12	Second formal meeting with instructor to discuss progress.
Dec 5	Complete survey paper due.
Dec 5 - Dec 8	Survey paper graded, start revisions.
Dec 14	Final project report and final survey paper due.
Dec 14-19	Final presentations.

Project Ideas

A *project* can take multiple forms and you have wide latitude in designing your specific project.

You are encouraged to come up with your own ideas for a project that is related to your research interests and discuss them with the instructor. This is your chance to take the time and get the support and even credit for a system project you've always wanted to do!

Your project can provide systems support for or solve a systems-related problem in a – preferably your own - research area. You may not merely develop an application.

For all projects, it is important that the focus lies on the systems aspects; if your project includes an application, it **must not be** the focus of the project. Projects that are purely application projects (e.g., a mobile app to access X) are not suitable for this course.

Here are some ideas of projects in research areas I am currently particularly interested in:

Support for Script Spaces in Google/Chrome.

Script Spaces are an abstraction that allow concurrent processing inside the confines of a single webpage. They can augment traditional multi-process approaches for browsers. See A. Deka's thesis for more information: http://scholar.lib.vt.edu/theses/available/etd-08132010-175539/

Requires coding in C++ and JavaScript.

A multi-core implementation of CloudBrowser.

CloudBrowser is an innovative development framework for web applications based on server-side processing of documents that are rendered client-side. The current version of cloud browser is restricted to only one node.js instance. To effectively support multiple cores, it needs to be extended so that connections can be handed off across node.js instances. In particular, the cross-process passing of websocket connections in socket.io must be implemented.

See McDaniel/Back 2012 for more information
http://theta.cs.vt.edu/~gback/splash2012/wf03-mcdaniel.pdf

Requires coding in C++ and JavaScript

Implementing CloudBrowser in tame.js

A traditional counter-argument against event-based programming has been that it requires stack-ripping, leading to what has been termed "callback spaghetti".

Tame.js is a system to avoid that. This project would port the CloudBrowser framework to use tame.js and perform a comparison (performance and design) of the two approaches.

Requires coding in JavaScript & possibly C++

Survey Paper Suggestions

Instead of merely summarizing a number of papers, a *survey paper* should explore a particular research area or controversy. It should include a historical perspective that explains the relevance of the chosen research topic. It should outline the design space, compare and contrast multiple different approaches to a problem and discuss their trade-offs.

Ideally, you would pick a survey paper in an area that is related to your research interests.

The following are some possible topics for a survey paper:

- *Threads vs. Events.* An old, unresolved controversy is which of these two models is better suited for designing concurrent systems. Since Lauer & Needham's paper in 1979, which was supposed to settle the issue, different researchers and systems designers have provided arguments and implementations favoring on or the other. Describe the controversy, and describe how changing assumptions and needs influenced the design of concurrent systems.
- *Hardware vs. Software Protection.* Traditional OS implement protection in hardware using distinct address spaces. During the last 15 years, several techniques have emerged that promise to implement at least some aspects of protection in software, such as software-fault isolation, type-safe languages, proof-carrying code. Compare and contrast these approaches.
- *Virtual Machines.* Virtual Machines, originally developed in the 70s, have recently seen a comeback. They are used for both desktop PCs and large servers. Survey the different techniques, paying particular attention to the degree to which hardware is virtualized and the trade-offs involved.

  Additional topics that have recently received interest in the OS community include

- User-level vs. kernel-level threading and hybrids
- Real-time scheduling techniques
- Ubiquitous/Pervasive computing
- Multi-tasking support for language-based virtual machines
- Applications of static program analysis to OS

- File Systems (metadata handling and recovery)
- Power and energy management
- New approaches to system administration and configuration
- Overlay networks
- OS support for homogeneous clusters
- OS support for multicore machines
- OS support for mobile devices
- Scalable servers (web servers, web caches)
- Structured peer-2-peer distributed systems

.