Project Guidelines and Schedule

Intro
You have the choice of either writing a survey paper or doing an implementation project. Survey papers are written individually whereas implementation projects can be done with a partner.

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

<table>
<thead>
<tr>
<th>Date</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sep 28</td>
<td>Initial proposal (max 2 pages) due.</td>
</tr>
<tr>
<td></td>
<td>For survey papers, submit a description that describes your initial research on the topic, including a list of literature you will study.</td>
</tr>
<tr>
<td></td>
<td>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 milestone 1 and 2 (see below.)</td>
</tr>
<tr>
<td>Sep 29-Oct 12</td>
<td>Work with instructor to revise and get approval for topic.</td>
</tr>
<tr>
<td>Oct 21</td>
<td>Survey: Outline of paper due. In addition, create and submit written summaries of the sources you are researching.</td>
</tr>
<tr>
<td>Oct 22-26</td>
<td>Project: Progress report 1. (1 page)</td>
</tr>
<tr>
<td>Nov 11</td>
<td>Survey: First complete draft of paper due.</td>
</tr>
<tr>
<td>Nov 12-18</td>
<td>Project: Progress report 2. (1 page)</td>
</tr>
<tr>
<td>Dec 7</td>
<td>Final survey paper or project report due.</td>
</tr>
<tr>
<td>Dec 13-17</td>
<td>Final presentation</td>
</tr>
</tbody>
</table>

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.

The following are some possible topics for a survey paper. You are encouraged to identify subjects that suit your interests besides these:

- **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.

- **Grid Computing.** The advent of high-speed networks, combined with cheaper, abundant processing power have changed the landscape of distributed computing. Explore the area of grid computing, paying particular attention to how distributed systems ideas and principles are applied in it and the requirements grid computing environments impose on the operating system.

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

- **Threads vs. Events.**
- **User-level vs. kernel-level threading and hybrids**
- **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**
Project Ideas

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

Following the dual structure of the course (a lecture track dealing mainly with distributed system techniques, and a presentation track with a variety of OS topics), these projects could involve distributed systems, but may also be related to other OS topics.

If you choose to implement a distributed system, you may use a variety of technologies, including but not limited to:

- Simple client/server communication
- Middleware such as CORBA
- Java-based systems such as RMI
- Tuple-based systems such as JavaSpaces/JINI
- .NET

Some ideas are listed below:

- Build a P2P system. Examples of tasks that could be implemented are: a file/storage system, a search engine, a chat system, a distributed game, a distributed webcache/content-distribution network (CDN) or a distributed computing environment (mini-grid) a la SETI@home. In addition to mastering the distributed nature of this problem, you should attack at least two of the following aspects:
  - Fault Tolerance. Include techniques, such as replication or erasure codes, that allow your system to recover from individual node failures.
  - Security. Provide security features, such as protection against unauthorized access, use of encryption to provide confidentiality, etc.
  - Scalability. Your algorithms to locate peers and data should scale as the number of peers grows, using such techniques as distributed hashtables (DHT).
- Distributed Implementation of JSR-121. The next version of Java will provide support for isolates, multiple applications
controlled by a single JVM. The Isolate API is designed to allow for multiple implementations (within a single OS process, using multiple processes and shared memory, etc.). Implement a distributed version of JSR-121 that allows Isolates to be located on different machines.

- OS support for Grids. Identify a grid service that is hampered by current OS shortcomings and devise a solution to improve this service. Examples might include resource reservations and cooperative scheduling for jobs.

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!