Graduate Operating Systems **Midterm Exam**
Prof. Steven Gribble **November 12, 2009**

CSEP551 Take-Home Midterm Exam

Due: Thursday November 19th, 2009, before class.

Guidelines: This exam is open book. You **may not** discuss it with anybody but the instructor/TA. If you use sources other than the papers that we have covered to date in class (e.g., other papers, Web pages, books, …), include a full bibliographic reference. We will answer questions through email or in person, and we will use the mailing list to broadcast any corrections.

Please submit your solutions using the following Catalyst drop-box:

https://catalysttools.washington.edu/collectit/dropbox/gribble/7604

There are a total of 4 questions in this exam, and a total of 120 points. Please read the questions carefully.

Each question includes a maximum number of pages you should use in your answer; feel free to use less space, but do not use more. Please use reasonable margins, font sizes and line spacing; if you cram, we'll be cranky.

Problem 1: Technology Trends [30 points]

Technology trends serve as an interesting lens through which we can examine systems. These trends typically dictate a system's architectural choices, design tradeoffs, and even its very feasibility.

(a) *(1.5 pages)* For each of the following, quantitatively describe its trend over the past 5 years. As well, extrapolate or use what we've learned in class to predict its trend over the next 5 years. Comment on cost, performance (bandwidth + latency), and/or capacity, where appropriate and meaningful.

- durable storage (SSDs or hard drives)
- transient storage (RAM)
- last mile networks
- processing capacity (speed, # cores, etc.)

(b) *(1.5 pages)* Discuss the implications of these trends on the feasibility and design of the following kinds of system. Focus on issues that were once important but are becoming less so, and designs/mechanisms that will be possible or required soon that were not in the past. Only some of the trends from (a) are relevant to each system…discuss only those that are.

- fault-tolerant, cloud-assisted file systems for home users

- non-monolithic operating system architectures, such as microkernels or multikernels

Problem 2: Grokking Graphs [30 points]

(3 pages) Tables and graphs in a paper serve many purposes. Most obviously, they present data collected from a system in a form that helps to answer some questions about the system. Second, they often expose interesting or unexpected phenomena about a system, or fundamental tradeoffs between system design and implementation choices.

For each of the following, identify and explain the "interesting" features that you find in the graph or table. Where possible, explain the technological underpinnings, workload aspects, or system tradeoffs and implementation choices that led to these features.

Some things you might want to look at are X and Y intercepts, slopes and knees in curves, unexpected bumps in the data, or comparisons/orderings/gaps between lines or data values.

(a) table 3 from *Extensibility, Safety and Performance in the SPIN Operating System*

(b) figure 7 from *The Multikernel: A new OS architecture for scalable multicore systems*

(c) figure 8 from *The Design and Implementation of a Log-Structured File System*

(d) figure 3c from *The Google File System*

(Note that the figure numbers are based off of the paper versions linked to from the course home page. Be careful to use these versions of the paper, rather than versions you download from other sources, so as to ensure you're analyzing the right figure/table.)

Problem 3: Quirky Questions [30 points]

a. *(1 page)* Assume that there is a test-and-set spinlock shared by multiple processes running on a multiprocessor machine. In "Thread Management Alternatives for Multiprocessors", we read about the Ethernet-style backoff algorithm. This algorithm has two serious negative properties: first, it assumes that all processes accessing the spinlock are well-behaved, i.e., they faithfully follow the exponential backoff algorithm. Second, it is unfair: processes that access a lock for the first time are more likely to get it than processes that have been waiting. Can either (or both) of these issues be remedied? If so, how?

b. *(1 page)* GFS does not provide strong consistency semantics to its applications. Depending on timing of messages, failures, and recoveries, different GFS clients might see different file state within GFS at the same time. We read about Paxos, which provides one way to build strong agreement into replicas, at the cost of performance and of liveness under some failure scenarios. Briefly how you would use Paxos to fix the consistency "problems" of GFS, and explain what you think the impact would be on GFS's performance and scalability.

Problem 4: Building Better Browsers [30 points]

(2 pages) In class, we've argued that Web browsers are becoming the new operating system. Browsers perform many of the same roles as OSs: they execute programs, they allocate and manage scarce resources, and they provide storage, communication, and protection abstractions to people, Web programs, and remote Web services. In the future, it is possible the Web browser and desktop OS will converge.

For each of the following, describe what choices today's browsers make, and propose better choices for the browser OS of tomorrow. Highlight issues that you believe are essentially the same for browsers of tomorrow and today's desktop OSs, as well as issues that will require a different approach.

- the "process" abstraction – i.e., the virtual machine in which programs execute
- extensibility – i.e., how web programs or users can modify the base mechanisms and abstractions that a browser provides
- inter-process communication – i.e., the manner in which web programs name and communicate with each other