CMPSC 441
Principles of Distributed Systems
Spring 2016

Examination 3 Study Guide
Delivered: Monday, May 2, 2016
Examination: Thursday, May 5, 2016, 7:00 pm

Introduction

This course will have its third examination on Thursday, May 5, 2016 from 7:00 to 10:00 pm. The examination will be "closed notes" and "closed book" and it will cover the following content. Please review the "Course Schedule" on the Web site for the course to see the content and slides that we have covered in these modules. You may post questions about this material to Slack.

- Chapters One through Two in DSPP (i.e., the fundamentals of distributed systems).
- Chapters Three through Four in DSPP (i.e., the use of processes and remote communication).
- Chapter Ten in DSPP (i.e., distributed object-based systems).
- Chapter Eleven in DSPP (i.e., distributed file systems).
- Chapter Twelve in DSPP (i.e., distributed web-based systems).
- Chapter Five in DSPP (i.e., naming in distributed systems).
- Chapter Eight in DSPP (i.e., fault tolerance in distributed systems).
- Your class notes, class activities, lecture slides, and all of the laboratory assignments.
- Knowledge of the basic commands necessary for using the Ubuntu operating system, `git` and Bitbucket; basic understanding of how to use LaTeX and Markdown for scientific writing.

The examination will include a mix of questions that will require you to draw and/or comment on a diagram, write a short answer, explain and/or write a source code segment, or give and comment on a list of concepts or points. The emphasis will be on the following list of illustrative subjects. Please note that this list is not exhaustive — rather it is designed to suggest representative topics.

- Fundamental Principles of Distributed Systems

– The state-of-the-art and the key challenges within the field of distributed systems, with a focus on the types of transparency and the challenges associated with their provision.
– The basic evaluation metrics for distributed systems (e.g., response time and throughput), how they would be calculated, and their overall relationship to properties like scalability.
– The architectural styles for implementing a distributed system. Specifically, an understanding of how to organize clients and servers to ensure, for instance, high reliability and performance.
– The trade-offs associated with the use of threads, processes, and light-weight processes in the context of clients and servers running in both centralized and distributed computing systems.
– The benefits and drawbacks from using resource virtualization in distributed systems.
– An understanding of the changing roles of clients and servers (e.g., in the X Window system).

- The trade-offs associated with designs for the servers (e.g., stateful versus stateless) and code migration techniques (e.g., weak mobility versus strong mobility) in a distributed system.
- The protocols commonly used in a distributed system (e.g., HTTP, FTP, and SFTP).
- All of the key steps associated with performing remote procedure calls (RPCs).
- The similarities and differences between communication with sockets and RPCs or RMIs.
- The different types of communication in a distributed system (e.g., multicast and unicast).

• Application Areas in Distributed Systems

- A basic understanding of object-oriented programming principles and the various ways in which objects are implemented and used in a distributed system (e.g., remote objects).
- The ways in which remote method invocations occur in a distributed object-based system.
- The trade-offs inherent in approaches to performing remote method calls (e.g., static vs. dynamic).
- The key features associated with distributed computing in the Java programming language.
- The upload/download and remote access model for accessing files in a distributed file system.
- The necessary details concerning the implementation and use of distributed file systems (e.g., a basic system architecture and a list of file system operations that must be supported).
- The ways in which a distributed file system could support distributed computing on a cluster.
- The implementation innovations necessitated by the creation of distributed file systems.
- The semantics for accessing files in distributed and local file systems (e.g., UNIX or session).
- All of the key terms associated with distributed web-based systems (e.g., "MIME type").
- Implementation details for both web servers and web clients (e.g., rendering and caching).
- The trade-offs inherent in creating web-based systems (e.g., static versus dynamic rendering).
- Communication techniques used in web-based distributed systems (e.g., HTTP operations).

• Advanced Topics in Distributed Systems

- Key terminology in distributed computing's use of names, identifiers, and addresses.
- The meaning of service identifiers that are either "location independent" or "true".
- The benefits and drawbacks of using multicasting to locate named entities.
- The forwarding-pointers and home-based methods for locating mobile entities.
- Key terminology associated with the implementation of structured namespaces.
- The iterative and recursive methods for name resolution in a distributed system.
- The basic concepts needed to understand fault tolerance in distributed computing.
- The different failure models associated with real-world distributed systems.
- An understanding of the methods for ensuring reliable and fault-tolerant distributed computing.

• General Concepts in Distributed Systems

- The principles for effectively benchmarking systems and reporting on performance results.
- The three different types of benchmarks commonly used in evaluating a system's performance.
- General strategies for implementing efficient distributed systems (e.g., "keep the computation close to the data") and the trade-offs inherent in designing distributed computing systems.
- Lessons learned from the laboratory assignments in which you implemented and experimentally evaluated distributed systems either individually or in a team-based setting.

Examination Policies

Minimal partial credit may be awarded for the questions that require a student to write a short answer. You are strongly encouraged to write short, precise, and correct responses to all of the questions. When you are taking the examination, you should do so as a "point maximizer" who first responds to the questions that you are most likely to answer correctly for full points. Please keep the time limitation in mind as you are absolutely required to submit the examination at the end of the test period unless you have written permission for extra time from a member of the Learning Commons. Students who do not submit their examination on time will have their overall point total reduced. Please see the course instructor if you have questions about any of these policies.

Review the Honor Code

Students are required to fully adhere to the Honor Code during the completion of this examination. More details about the Allegheny College Honor Code are provided on the syllabus. Students are strongly encouraged to carefully review the full statement of the Honor Code before taking this test.

The following provides you with a review of the Honor Code statement from the course syllabus:

The Academic Honor Program that governs the entire academic program at Allegheny College is described in the Allegheny Academic Bulletin. The Honor Program applies to all work that is submitted for academic credit or to meet non-credit requirements for graduation at Allegheny College. This includes all work assigned for this class (e.g., examinations, laboratory assignments, and the final project). All students who have enrolled in the College will work under the Honor Program. Each student who has matriculated at the College has acknowledged the following pledge:

> I hereby recognize and pledge to fulfill my responsibilities, as defined in the Honor Code, and to maintain the integrity of both myself and the College community as a whole.

Students who have questions about the College's Honor Code and how it applies to the completion of an examination in this course should immediately schedule a meeting with the instructor.

Strategies for Studying

As you study for this examination, you are encouraged to form study groups with individuals who were previously, during a laboratory session, a member of one of your software development and empirical study teams. You can collaborate with these individuals to ensure that you understand all of the key concepts mentioned on this study guide. Additionally, students are encouraged to create a Slack channel that can host questions and answers that arise as you continue to study for the test. Even though the course instructor will try to, whenever possible, answer review questions that students post in this channel, you are strongly encouraged to answer the questions posted by your colleagues as this will also help you to ensure that you fully understand the material.

When studying for the final examination, don't forget that the Web site for our course contains mobile-ready slides that will provide you with an overview of the key concepts that we discussed in all of the modules for this course. You can use the color scheme in the slides to notice points where we, for instance, completed an in-class activity, discussed a key point, or made reference to additional details available in the DSPP textbook. Finally, students should carefully review all of the assignment sheets and their graded laboratory and final project submissions so that they better understand all of the key concepts that we examined in each component of the course.