SE 3SH3 Operating Systems

Rong Zheng

February 14, 2015

1 Precondition

Students are expected to have knowledge in computer architecture, and data structures such as (double) linked list, hash table, array. Students should have prior experiences with code tracing, Java programming and Unix commands.

2 What the students should know and be able to do

1. Students should know and understand
   (a) evolution of operating systems, the concepts and techniques developed.
   (b) concepts of processes and threads.
   (c) synchronization mechanisms, semaphore, lock, and condition variables.
   (d) design and implementation principles of the file system in a modern operating system.
   (e) design and implementation principles of memory management in a modern operating system.
   (f) design and implementation principles of CPU scheduler in a modern operating system.
   (g) design principles of distributed systems.
2. Students should be able to
   (a) write well structured and well organized programs.
   (b) write multi-threading programs.
   (c) write programs applying synchronization mechanisms.
   (d) design and implement priority scheduling.
   (e) design and implement a file system interface.

(f) design and implement virtual memory.

(g) design and implement testing programs

(h) document code.

3 Mapping to Attributes with Their Indicators

A01 Knowledge	
Competence in specialized engineering knowledge	1a–1g
A03 Investigation	
Uses appropriate techniques to collect data	2g
Assess the accuracy and precision of results and recognize limitations of the approach	2g
A04 Design	
Recognizes and follows an engineering design process	2b–2h
Recognizes and follows engineering design principles	2b–2g
Obtains experience with open-ended problems	2b–2f
Properly documents and communicates processes and outcomes	2h

Table 1: Rubric: (4) **Competence in Specialized Engineering Knowledge**

Topic	**Below**	**Marginal**	**Meets**	**Exceeds**
Operating system evolution 1a	no knowledge of basic concepts in operating systems	basic knowledge of concepts in operating systems	good knowledge of concepts in operating systems	good knowledge of concepts in operating systems and examples
Processes and threads 1b	no knowledge of the concepts of processes and threads	basic knowledge of the concepts of processes and threads with difficulty in applying them	basic knowledge of the concepts of processes and threads and their applications	good knowledge of the concepts of processes and threads and their applications
Synchronization 1c	no knowledge of synchronization mechanisms such as semaphores and locks and conditions	basic knowledge of synchronization mechanisms with difficulty in applying them	basic knowledge of synchronization mechanisms and their applications	good understanding of synchronization mechanisms and their applications
File systems 1d	no knowledge of file system	basic knowledge of file system in a modern operating system	basic knowledge of file system in a modern operating system and its operations	good understanding of file system, its operations, and design principles

Table 2: Rubric: (4) **Competence in Specialized Engineering Knowledge**
Student work used: midterm and final exams. Each topic will be covered by an exam question.

Topic	**Below**	**Marginal**	**Meets**	**Exceeds**
Memory management 1e	no knowledge of memory management	basic knowledge of memory management, virtual memory, paging	basic knowledge of memory management and its functions	good understanding of memory management, its operations, and design principles
CPU scheduling 1f	no knowledge of scheduling (algorithms and performance measurements)	basic knowledge of scheduling in a modern operating system	basic knowledge of scheduling and its algorithms	good understanding of scheduling algorithms and their impact on performance
Distributed systems 1g	no knowledge of distributed system component (layering structure, protocols)	basic knowledge of distributed system component	basic knowledge of distributed system component and its functions	good understanding of distributed system component, its functions, and design principles

Table 3: Rubric: **What students should be able to do**
Student work used: Assignments 1 to 3

Topic	**Below**	**Marginal**	**Meets**	**Exceeds**
Threads and synchronization 2a–2c, 2g, 2h A1 problem: Implement and apply synchronization mechanisms	unable to design and implement synchronization mechanisms	basic design and implementations of synchronization mechanisms but unable to apply them	basic design and implementations of synchronization mechanisms and their application with difficulty in testing	good design and implementations of synchronization mechanisms and their application with thorough testing
CPU scheduling 2a–2d, 2g, 2h A2.1 problem: Design and implement priority scheduling using	no knowledge of CPU scheduling	basic understanding of CPU scheduling but unable to implement priority scheduling	good understanding of priority scheduling and basic design and implementation of an priority scheduling	good understanding of priority scheduling and good design and implementation of an priority scheduling thorough testing

Table 4: Rubric: **What students should be able to do**
Student work used: Assignments 1 to 3

Topic	**Below**	**Marginal**	**Meets**	**Exceeds**
File system 2a–**??**,2g,2h A2.2 and A3.1 problem: Design and implement file system calls and main memory management	no understanding of file systems and its functions	basic understanding of file systems and its functions but unable to implement it	good understanding of file systems and its functions and basic design and implementation of the component	good understanding of file systems and its functions and good design and implementation of file system calls thorough testing
Virtual memory 2a–**??**,2g,2h A3.2 problem: Design and implement virtual memory management	no understanding of virtual memory management and its functions	basic understanding of virtual memory management and its functions but unable to implement it	good understanding of virtual memory management and its functions and basic design and implementation of virtual memory management	good understanding of virtual memory management and its functions and good design and implementation of virtual memory management with thorough testing