THE IMPACT OF E-COMMERCE ON DEVELOPING A COURSE IN OPERATING SYSTEMS: AN INTERPRETIVE STUDY

Reggie Davidrajuh, Stavanger University College, Norway, reggie.davidrajuh@tn.his.no

ABSTRACT

This paper presents a goal-oriented top-down design approach for development of a new course in Operating Systems as a part of Masters degree in E-commerce. We developed this course by going through e-commerce infrastructures and architectures.

Keywords: Operating systems, e-commerce, information technology education

INTRODUCTION

Operating systems are an essential part of any e-commerce infrastructure. Operating system (OS) is the most important program that runs on a computer. Thus, a course in OS is an essential part of any computer science or information technology (IT) education. OS have been continuously evolving ever since they were introduced in the mid-1950s for 2nd generation computers, to present-day e-commerce era. In this paper we present the preliminaries of developing a new course in OS that would be an essential part of a Masters degree in IT/E-commerce.

Overview of Courses in Operating System

OS defines an abstraction of hardware behavior with which programmers can control the hardware; OS also manages resource sharing among the computer's users (2). The traditional courses in OS include a practical (lab) module and a theory module (3); the practical module is to enable students to experiment with OS use (externals) and the theory module addresses design and implementation details (internals). Courses in OS are normally given at two levels: Bachelors level basic course and Masters level advanced course.

Why a New Course in Operating System?

Present-day courses assume that the primary job of an OS is to assist application programs by acting as an intermediary between application programs and the hardware; this is the view of OS that were suitable for the 4th generation computer environment.

We consider e-commerce as a means of merging business logics of different enterprises that undergo partnership. Consequently, modern day OS should be viewed as an intermediary to achieve inter and intra enterprise data migration, computation migration, and process migration. We believe the courses in OS should be designed on this basis.

Overview of this paper

This paper presents a unique design approach for development of a course in OS for an E-commerce degree. During the course development, we place an IT specialist (or CIO, CTO) in a real-life e-commerce environment, and think about what kind of problem she will experience; what kind of knowledge she needs to select, maintain, and upgrade e-commerce infrastructure.

This type of goal-oriented top-down design approach enables us to develop a course that is not only relevant to the industry but also interesting to the students.

In the next section, we present contents of the contemporary courses in OS. In the third section, we present our proposal for a new course in OS.

EXISTING COURSES IN OPERATING SYSTEMS

In the following subsections, we present the topics that are discussed in the present-day courses (basic and advanced) in OS.

Topic	Contents
1. Overview	What operating systems are, what they do, and how they are designed and constructed.
2. Process management	At the heart of modern operating systems are processes, the units of work, which execute concurrently. The main issues dealing with processes are process scheduling, inter-process communication, process synchronization, and deadlock handling.
3. Memory management	The processes must be in main memory during execution. The size of the main memory is too small compared with the size of processes. The main issues dealing with memory management are main-memory allocation, swapping, overlays, virtual memory, and policies and algorithms for page-replacement.
4. Input/Output management	One of the main functions of an operating system is to manage and control the computer's input/output (I/O) operations and I/O devices. Some of the issues raised in this section are, I/O hardware types, application I/O interface, and I/O requests and handling.
5. File management	Since the main memory is too small and volatile to permanently accommodate all data and program, the computer system must provide secondary storage (e.g. hard disks) to backup main memory. The main issues dealing with file systems management are file naming, file structure and file implementation issues.
Table-1: Topics in operating systems courses	

Basic Course in Operating System
Basic courses address stand-alone computer systems (without consideration to network). The practical module deals with practical exercises on Windows NT, Linux or UNIX based workstations. The theory module covers topics such as processes management, main memory management, management of input-output devices, and file systems management (4, 8); see table-1 for details. Many universities (e.g. 9, 10) offer a basic course in OS with contents similar to that in table-1.

Advanced Course in Operating System
Over the years, OS and their abstractions have become complex relative to typical application software. Therefore it has become necessary to ensure that the student study advanced OS topics and be exposed to recent developments in OS research.

An advanced course in OS generally introduces the concepts that are fundamental for understanding distributed systems and the technical infrastructure that makes them possible. The course examines why the issues that are presented in the basic course in OS, like concurrency, mutual exclusion, deadlocks, and scheduling, would partially fail in distributed systems, mainly due to lack of a single clock.

The theory module may cover topics such as network structures, distributed communication and coordination, distributed file systems, and protection and security (1, 4); see table-2 for details. Rather than using standard textbooks, the primary readings in advanced courses are normally supplemental research papers (e.g. 9).   The lab module may involve programming projects for distributed applications.

Topic	Contents
1. Review of operating systems	Review of operating systems and systems programming concepts including Inter-Process Communications and the use of Threads. Other issues raised under this topic are the Client-Server model and Performance Evaluation.
2. Networks and data communications	Review the networks that make distributed systems possible. The main issues under this topic are architectures, TCP/IP suite, socket programming, and Remote Procedure Calls, and Remote Method Invocation.
3. Distributed coordination	Description of various mechanisms that allow processes in a distributed environment to synchronize their actions. The main issues are event ordering, mutual exclusion and deadlock handling in a distributed environment, main-memory allocation.
4. Distributed file systems	This topic is about implementing and supporting file systems sharing when the files are physically dispersed among various sites. Some of the issues raised under this topic are, naming and transparency, remote file access, and file replication.
5. Protection and security	Protection mechanisms ensure that only the authorized processes use resources defined by a computer system. Security mechanisms prevent unauthorized access to a system. The main issues are access matrix, revocation of rights, language-based protection, authentication, encryption, and computer-security classifications.
Table-2: Topics in an advanced course in operating systems	

A NEW COURSE IN OPERATING SYSTEM FOR DEGREE IN E-COMMERCE

The author of this paper is involved in developing a new course in OS, called Applied Operating Systems (AOS). This course is intended for students following Masters degree in IT/E-commerce. Before we present the course contents of AOS in subsection 3.3, we formulate the design criteria for the new course; the formulation is done in the next two subsections on: 1) components of e-commerce infrastructure 2) E-commerce architecture that facilitates e-commerce applications.

Components of e-commerce infrastructure

Figure-1 shows the major components of e-commerce infrastructure that provide the foundation for the elements of e-commerce architecture; e-commerce architecture is presented in the following subsection. Basically, figure-1 shows the major components of e-commerce infrastructure that enable data migration, computation migration, and process migration in a heterogeneous environment. Enterprises often employ more than one OS owing the reasons such as mergers, new application requirements, etc.

In addition to the hardware components, figure-1 identifies four important software components: 1) Basic, and 2) network OS services: The basic OS issues are presented in table-1. Network OS issues include remote login, remote file systems and file transfer, and file transfer protocol (FTP) for data migration, etc; network OS issues are given in table-2.

3) Protocols: Internet Protocol (IP), Transmission Control Protocol (TCP), and User Datagram Protocol (UDP) are some of the principal protocols for communication across Internet. IP is used for transmitting datagrams, the basic unit of information; computation migration happens (to be specific, remote procedure call RPC) via UDP; TCP uses IP to transport a reliable stream of data between two processes (5).

4) Middleware: Since enterprises employ different hardware and OS, we need special software called middleware to obtain interoperability between heterogeneous systems. Middleware can be generally grouped into three categories: communication (e.g. Message-Oriented Middleware - MOM), data management (e.g. Extended Markup Language - XML), and platform middleware (e.g. Common Object Request Broker Architecture - CORBA) (13).
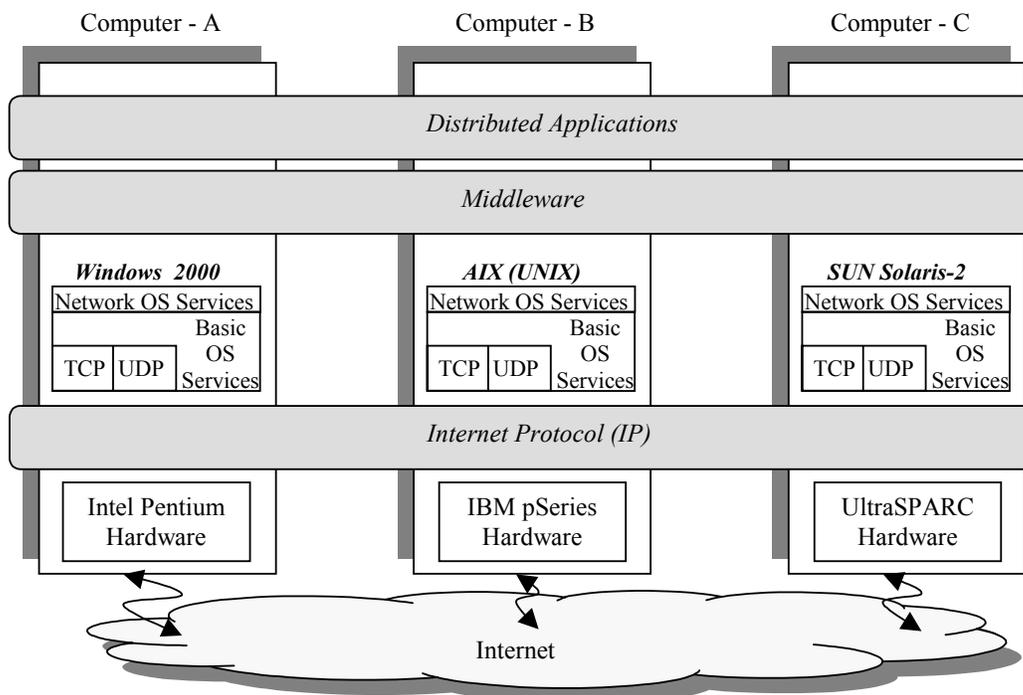
Figure-1: Major components of e-commerce infrastructure.

5) Distributed Applications: Distributed applications means applications that can be run across network, which access remote resources in the same manner as in the case of local applications. Process migration that offers advantages such as load balancing, computation speedup, etc., comes under distributed application paradigm.
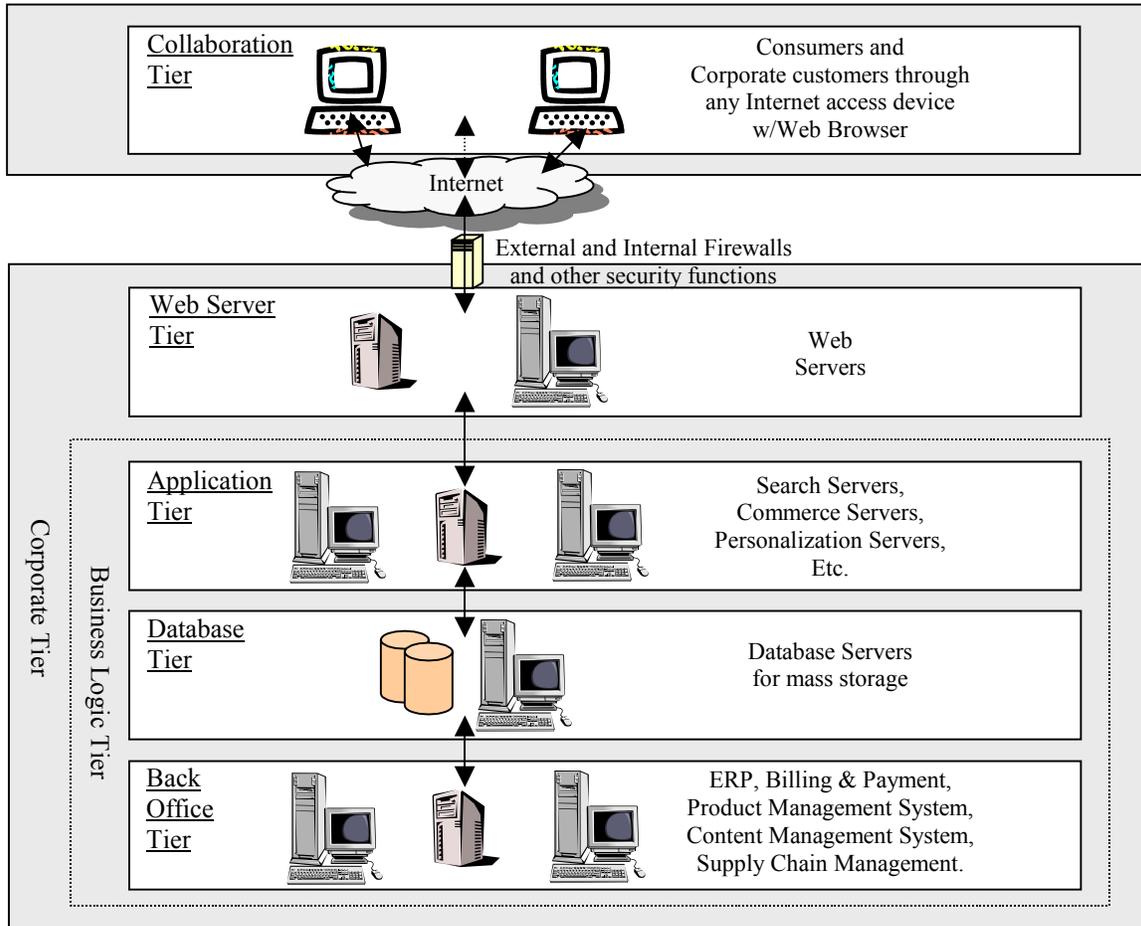

Figure-2: n-tier e-commerce architecture

E-commerce Architecture
Architecture for B2B/B2C e-commerce that allows modularity to ease functional extensions, scalability for future growth, and high reliability (fault-tolerant), is shown in figure-2 (6, 13). E-commerce architecture shown in figure-2 identifies many elements. There will be at least 3 common elements in any e-commerce architecture:

1) Firewalls and other security functions: Security (authentication, encryption, etc.), privacy and protection (access rights and privileges) are important elements of e-commerce architecture.
2) Web server: The role of the Web server is to host Web pages. One of the key features to look for in a Web server is the speed of processing ('response time'). One of the popular Web servers is iPlanet, primarily designed to work with Solaris OS on Sun SPARC workstations; the other one is the Apache, used together with Linux OS.

3) Database server: A crucial element of e-commerce architecture is databases and database server. In selecting database server hardware, storage capacity is one of the key factors. In selecting the database server software, the key requirements are scalability, multiple access, and support for data integrity (ensuring the data in the databases are consistent, accurate, and valid) (13).

AOS - the New Course in Operating System
In the previous subsection (3.1 and 3.2), we have seen the major components of e-commerce infrastructure and the essential elements of e-commerce architecture. Basic knowledge about these components is obligatory for any IT/e-commerce technologist. Based on the components identified in the previous subsections, we present our proposal for the contents of the new course. Structure of the new course (AOS) is shown in figure-3.
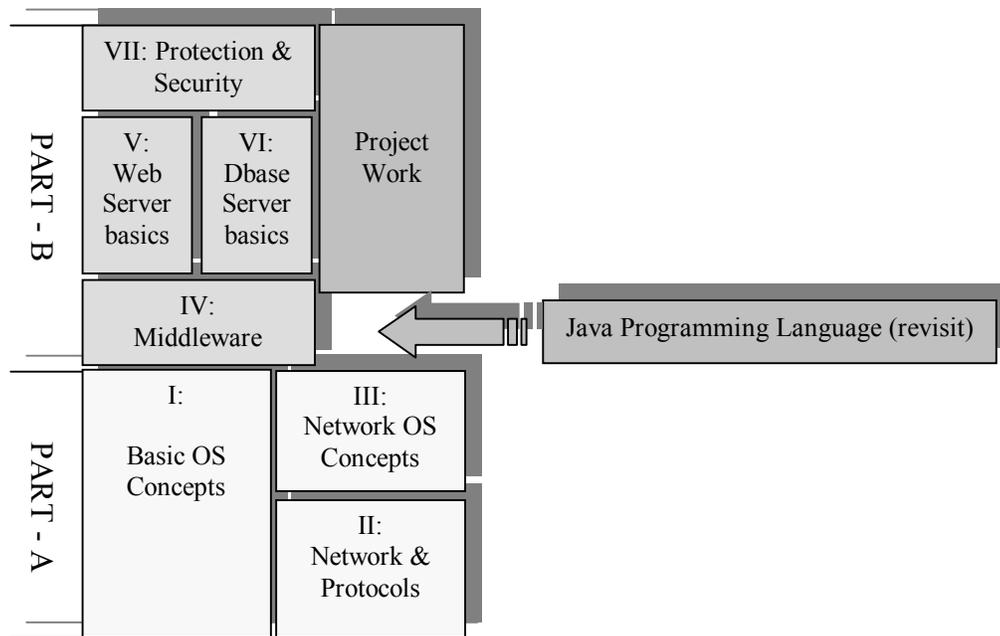

Figure-3: Structure of AOS

AOS is divided into two main parts (A and B), each carry 10 study points (10 SP is equivalent to 6 lecture hours and 3 exercise and group work hours for 12 weeks). Part - A is further divided into 3 modules. Module-I deals with basic OS concepts like the ones listed in table-1. Module-II deals with basic networking issues like topologies, technologies, and protocols (like IP, TCP, UDP, and Hypertext Transport Protocol - HTTP). Module-III deals with network OS concepts like socket programming principles, Remote Procedure Calls, Remote Method Invocation, and distributed file systems.

Part - B is further divided into 4 modules (modules IV – VII). Module-IV is about middleware; different types, models, architectures, and services of middleware are studied. Module-V deals with basic Web server issues; learning objectives in this module are designing a web site,

understanding principles of a Web site management, understanding how a Web browser can access a database server, and likely developments in Web technology and applications. Module-VI is about basic database server issues such as transparency, distributed query and transaction processing, replication, fragmentation, hardware network and OS independence, local autonomy etc. Module-VII deals with protection and security issues such as firewalls, authentication, encryption, viruses, worms, etc. Finally, the practical module (project work) is programming lightweight distributed applications in client-server systems, using Java programming language. A short course (or revision) in Java language may be needed for those who have not used Java before but familiar with other programming languages like C++.

CONCLUSION

This paper presents a goal-oriented top-down design approach for development of a new course in Operating Systems as a part of Masters degree E-commerce. We developed this course by after studying e-commerce infrastructures and architectures. The new course has two main parts (A and B). Part - A is further divided into 3 modules, such as basic OS concepts, basic network principles and protocols, and network OS concepts. Part - B is further divided into 4 modules, such as middleware, basic Web server concepts, basic database server concepts, protection and security concepts, and finally, a practical module (project work) in programming distributed applications.

REFERENCES

1. Chow, R. and Johnson, T. (1997). Distributed Operating Systems and Algorithm Analysis, Addison-Wesley.
2. Davidrajuh, R. (2002). Lecture notes in Operating Systems, http://ied.ux.his.no/main.cfm?page=course&ID=387
3. Davidrajuh, R. (2003). A low-cost solution for computer-supported collaborative learning, Global Information Technology Management (GITM) World Conference, Calgary, Canada.
4. Silberschatz, A., Galvin, P., and Gagne, G. (2000). Applied Operating Systems Concepts, Wiley.
5. Silberschatz, A., Galvin, P., and Gagne, G. (2002). Operating System Concepts, 6th ed., Wiley.
6. Sun Microsystems. (2003). E-Commerce: Success Stories – Checkout.com www.sun.com/solutions/third-party/ecommerce/success/.
7. Tanenbaum, A. S. and Van Steen, M. (2002). Distributed Systems: Principles and Paradigms, Prentice-Hall, NJ.
8. Tanenbaum, A. S. and Woodhull, A. S. (1997). Operating Systems: Design and Implementation, 2ed., Prentice-Hall, NJ.
9. Univ. California, Santa Cruz. (2003). www.cse.ucsc.edu/~sbrandt/courses/Spring02/111/
10. University of Wisconsin, Madison. (2003). www.cs.wisc.edu/~solomon/cs537/cs537.html
11. University of Southern California. (2003). www.isi.edu/gost/courses/USC_CSci555.html
12. University of California, Berkeley. (2003). www.cs.berkeley.edu/~adj/cs262/
13. Van Slyke, C. and Belanger, F. (2003). E-business Technologies, Wiley, NY