

The Van Houten library has removed some of the personal information and all signatures from the approval page and biographical sketches of theses and dissertations in order to protect the identity of NJIT graduates and faculty.

Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be “used for any purpose other than private study, scholarship, or research.” If a user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of “fair use” that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select “Pages from: first page # to: last page #” on the print dialog screen

ABSTRACT

TEACHING OPERATING SYSTEM CONCEPTS USING MULTIMEDIA AND INTERNET

**by
Amit Narielwala**

The prime objective of the thesis is to research and demonstrate the benefits and advantages of using Internet and multimedia tools for an interactive educational learning experience. As we speak Internet is developing as a mainstream communication medium via personal computer as a tool at a breathtaking speed. The information technology field is a prime reason behind such phenomenon as it continues to mature and expand. In what is described as the "information age", the students of information technology need to master and devour new complex technological concepts and ideas at faster rate than ever before. The traditional approach using the textbooks is not feasible due to their static, linear and often colorless nature. So there is a tremendous need to develop an interactive, fun and yet detailed and challenging educational experience.

In next several chapters, the solution is presented as to how to tackle such a challenge or task, by using the Operating System concepts, specifically using Memory Management concepts as a case study. The concepts include topics such as logical vs. physical address space, etc.

**TEACHING OPERATING SYSTEM CONCEPTS
USING MULTIMEDIA AND INTERNET**

by
Amit Narielwala

**A Master's Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial fulfillment of the Requirements for the Degree of
Masters of Science in Computer Science**

Department of Computer and Information Science

August 1999

Blank Page

APPROVAL PAGE

**TEACHING OPERATING SYSTEM CONCEPTS
USING MULTIMEDIA AND INTERNET**

Amit Narielwala

Dr. Franz Kurfess, Thesis Advisor Date
Professor of Computer and Information Science, NJIT

Dr. James McHugh Date
Associate Chairperson of Computer and Information Science, NJIT

Mr. Leon Jololian Date
Professor of Computer and Information Science, NJIT

BIOGRAPHICAL SKETCH

Author: Amit Narielwala
Degree: Master of Science in Computer Science
Date: August 1999

Undergraduate and Graduate Education:

- Master of Science in Computer Science
New Jersey Institute of Technology, Newark, NJ 1999
- Bachelor of Science in Civil Engineering,
Rutgers University, New Brunswick, NJ 1996

Major: Civil Engineering

To my lovely family

ACKNOWLEDGMENT

I would like to thank Professor Franz Kurfess for his timely advise and all the students who worked with me on the operating system project.

TABLE OF CONTENTS

Chapter	Page
1 INTRODUCTION	1
1.1 Objective.....	1
1.2 Problem Statement.....	2
1.3 Background Information.....	3
2 REVIEWING TOOLS.....	4
3 IMPLEMENTATION.....	7
4 USER INTERFACE SPECIFICATIONS.....	14
4.1 Introduction.....	14
4.2 Design Considerations.....	16
4.3 Principles of Interface Design.....	17
4.4 User Classification.....	19
4.5 Design Process.....	22
4.6 Evaluation and Testing of the System.....	24
5 CONCLUSIONS.....	26
APPENDIX A USER MANUAL.....	28
APPENDIX B JAVA APPLET CLASS DEFINITIONS.....	37
REFERENCES.....	41

LIST OF FIGURES

Figure		Page
3.1	System's organization for maximum interactivity.....	7
3.2	Flow of the hyperlinked interactive websites and their content.....	8
3.3	Snapshot of an applet interactively explaining contiguous allocation.....	10
3.4	Snapshot of an applet interactively explaining logical and physical address.	11
3.5	Snapshot of an interactive quiz applet.....	12
3.6	Snapshot of an interactive quiz created by Shockwave.....	13
4.1.1	Various disciplines	15

CHAPTER 1

INTRODUCTION

1.1 Objective

The purpose of this paper is to outline the development of tool-sets to improve the teaching of operating system concepts. People retain about 20% of what they see, 30% of what they hear, 50% of what they see and hear, and up to 80% of what they see, hear, and do simultaneously. The usage of Internet and multimedia methods for the explanation, demonstration, visualization, animation, and interactive exploration of important operating system concepts will maximize a student's retention. And thus overcoming the restrictions of explaining difficult and abstract concepts. The advantage of these tools is that they can be used in the classroom as well as in distance learning which are gaining more popularity day by day. In order for a complete educational or learning experience, the following five techniques can be used:

- Explanation: Use hypertext to explain the topic tailored for the students depending on their skill level.
- Visualization: Use of diagrams or pictures to accompany text-based explanations of concepts
- Demonstration: Use dynamic demonstrations of methods or processes where pictures not suffice
- Animation: Use animated display of methods and processes for complex and lengthy concepts

- Interactive Exploration: Use of interaction for testing, applying, getting immediate feedback of the learned material

All of these concepts are explained and covered in detail in the case study to solidify the argument of using multimedia and Internet in teaching operating system concepts. The case study uses the memory management concept in operating system because it is an abstract, complex and dynamic concept.

1.2 Problem Statement

Textbooks remain the major source of educational tool today in our educational institutions. Unfortunately, the textbook approach has many weaknesses to be used a sound educational tool. Most books on operating system are strong on theory and weak on practice. So many students new to the computer science field with very little hands on or practical experience are left with distorted view of subject despite studying and knowing Operating System concept theories such as deadlocks, file management systems, etc. Most books and courses spend an enormous amount of time to scheduling algorithms, for example, which in practice are usually about a page of code, while completely ignoring I/O, which is often 30 percent of system or more. Also, many books use haphazard approach to describe the topics for example, starting with easy concepts and then explaining difficult concepts. I think that students gain a better understanding of complex material if it is represented in a coherent manner. So one way to combat the problem us to teach or present material is by describing the construction of an operating system from the bottom up, starting at the interface with the machine hardware and ending at the interface with the user.

By the end of material or lesson, the system has been constructed is seen to possess the features demanded at the beginning, attention to structure and logical dependence are the best means we have of building operating systems which are easy to understand, easy to maintain and relatively error free.

Another problem with the textbooks is in incoherent restriction in the book format: Concepts involving changes over time, e.g. different states of objects which involve movements and modifications, cannot be represented adequately on paper. Another shortcoming of a textbook is its static and sequential nature of the format since many computer architecture concepts are dynamic. Lack of interactivity also plays a major role in prohibiting a student's grasp of learning because it doesn't allow the student to test or inquire the study material thus making it possible to quickly forget.

1.3 Background Information

Multimedia usually means events that include some combination of sound (music and/or voice), still image (scanned and/or synthesized computer graphics), motion image (computer animation, cinema, and/or video) and text. (Curiously, multimedia historically meant slide shows with sound.) Interactive means that the user/browser/audience has the ability to act to influence the flow of events or to modify their form. The term "hyper" in hypermedia usually is used to mean a particular kind of non-linear, flexible interactive structure with built-in linking capabilities in which viewers can each choose their own path through some material. This article will demonstrate that it is useful to begin to differentiate the very wide range of activities can fit under this umbrella definition of interactivity.

CHAPTER 2

REVIEWING TOOLS

As more and more technologies are available day by day, it gets harder to review them. Following reviews describe strengths of different tools and explain why a particular tool was chosen.

HTML (Hyper Text Markup Language)

Hypertext, the non-linear medium, are electronic texts embedded with links to other texts breaking down the traditional linear narrative of the written word by encouraging readers/users to find their own paths. It was the first tool to enhance interactivity on the net and is the native data format of the Internet. Dynamic HTML is a collective term for a combination of new Hypertext Markup Language (HTML) tags and options, style sheets, and programming that creates Web pages more animated and more responsive to user interaction than previous versions of HTML. Dynamic HTML can allow Web documents to look and act like multimedia productions.

Java

In a perfect world, programming languages wouldn't matter to end-users. After all, how often do you stop to think about what language was used to create Microsoft Word? In the real world, however, programming languages do matter--they have a profound effect on product features, performance, and the costs of hardware and administration.

Java offers a solution to this problem by isolating applications from both hardware and operating systems. Unlike Windows programs, which are almost always compiled for that platform using tools that preclude their running on anything else, Java applications can be designed to run on any system that has a Java Virtual Machine (VM), also known as a Java Runtime Engine. The VM takes Java byte code and transforms it on the fly into instructions that can be executed by the PC. The same byte code can be interpreted as hardware-level instructions for any other platform with a VM implementation. So for an educational institution, Java makes perfect sense due to the existence of multiple platforms in computer labs.

Java also allows client-side programming via the applet. Applets rely on a Web browser with a VM (such as Microsoft Internet Explorer or Netscape Communicator) and are designed to be downloaded. Students, who are connecting from home or using the school computer lab, can download an applet and run it again and again like it was stored locally in the computer. The small bandwidth or the slow connection speed problem is eliminated. So Java applets reduce the server hits and at the same time allow the clients to do most of the processing.

Java Script

Java Scripts are client-side scripts used mainly for small processing tasks on the clients' machine making it fast and reducing the load on the server. Mostly used for error-checking and small input screens. Java script is preferable over VB script because it runs on various platforms such as Windows and Unix while VB script runs only on the Windows platform.

CGI Script

CGI script are server-side scripts due to their ability to dynamically create html pages based on the user's input and thus making it very interactive. They can be written either in Perl or C. The alternative to cgi scripts was to choose Active Server Page(ASP) which is very similar to CGI but instead it runs only on the Microsoft Internet Information Server(IIS).

Macromedia's Shockwave

Macromedia's Shockwave contains a built-in support for multimedia hyperlinks and hypertext for creating intelligent navigation. With one simple command, all supported media types such as text, graphics, sound, video, and animation can be linked and navigated. It supports standard Internet protocol delivering rich, interactive content anywhere on web. It allows the Web Master to integrate existing content with HTML, to distribute media anywhere with HTTP, and to integrate two-way communication (CGI post).

Apple's QuickTime

Quicktime is a multimedia development, storage, and playback technology from Apple. Quicktime files combine sound, text, animation, and video in a single file. It is supported on Mac and Windows platform.

CHAPTER 3

IMPLEMENTATION

In chapter 2, different technologies were reviewed for our use. These tools were used to create an interactive experience for the user/student. The following components are utilized and organized cohesively to make the learning experience interactive:

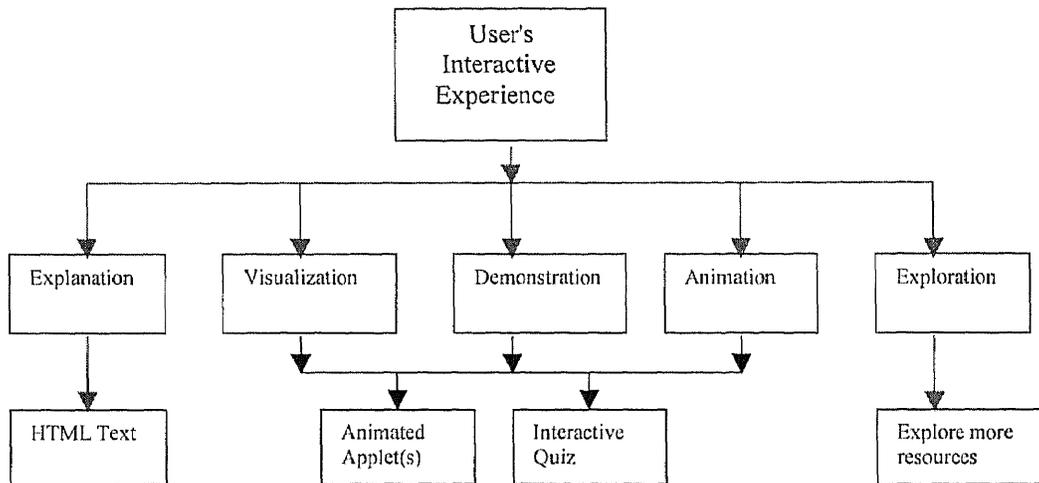


Figure 3.1 System's organization for maximum interactivity

The interactive website contained the components described in figure 3.2. It depicts the possible flow of activities by using the arrows.

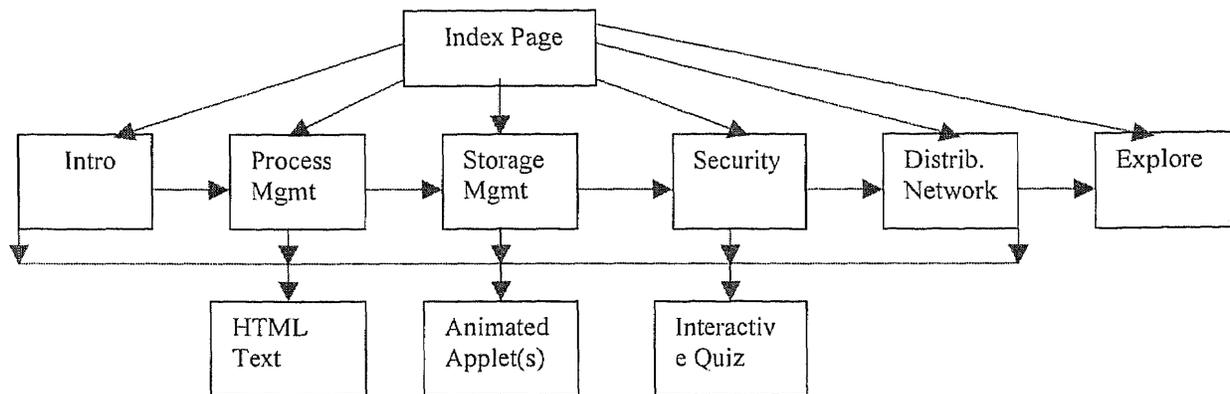


Figure 3.2 Flow of the hyperlinked interactive websites and their content

Java Development Kit version 1.1.8 was used to develop the applets animating the memory management concepts. A process must be in main memory during execution. To improve both the utilization of CPU and the speed of its response to its users, the computer must keep several processes in memory or manage memory. Memory management is a general term that covers all the various techniques by which an address generated by a CPU is translated into the actual address of the data in memory. Memory management plays several roles in a computer system. First, memory management permits computers with small main stores to execute programs that are far larger than the main store. Second, memory management is used in multitasking operating systems to make it look as if each task has sole control of the CPU. Third, memory management can be employed to protect one task from being corrupted by another task. Finally, memory management, in conjunction with the

operating system, deals with the allocation of memory to variables. The memory management discussion involves topics such as logical vs. physical address space, swapping, contiguous allocation, paging, segmentation, demand paging, page replacement, thrashing, etc. These topics are complex, abstract, and new to the undergraduate students of the operating system class. Next few topics discuss the issues, suggestions, and solutions implemented dealing with the complex topics.

The snapshots of the interactive applets were taken of the website created for the purposes of this project. The interactive website can be viewed at <http://unity.njit.edu/OS>. Along with interactive quizzes, animated applets, and textual definitions, it contains numerous links allowing the student to explore further beyond the scope of the textbook.

Logical vs. Physical Address Space

Logical address, an address generated by the CPU, is part of a set called logical address space which is generated by a program. The memory-management unit (MMU), a hardware device, maps these logical or virtual addresses to physical addresses based on the relocation register during run-time. A java applet showing the simulation of this mapping makes such an abstract concept very easy to grasp. From the picture below, which can be used as the interface on the animation, the user realizes that there is no actual logical address. The student/user can understand that the logical address is used as a frame of reference with the relocation register to get the actual physical address occupying in the hard disk. The applet in figure 3.1 was developed using JDK 1.1.8.

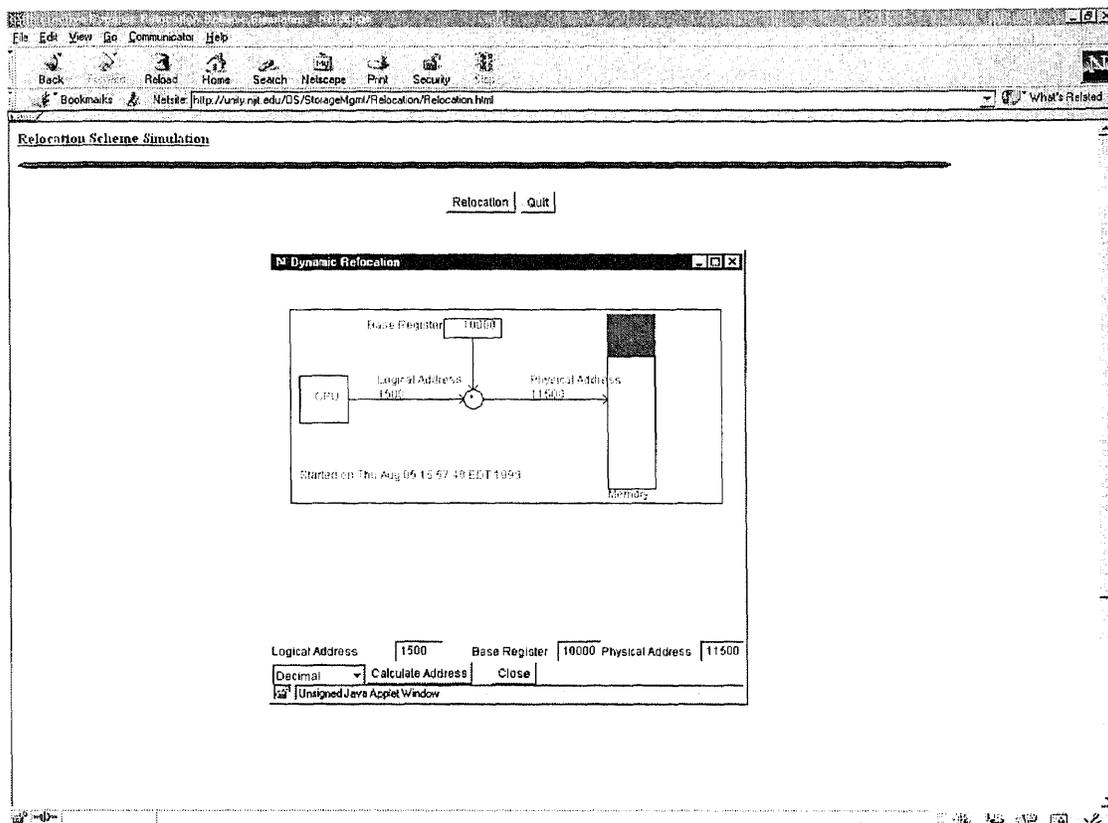


Figure 3.3 Snapshot of an applet interactively explaining logical and physical address

Contiguous Allocation

Contiguous allocation concepts involve understanding the purpose of limit register and relocation register. Contiguous allocation method requires each file to occupy a set of contiguous blocks on the disk. The addresses are ordered in linear manner. A screen shot of an animated applet showing the checks done by the operating system for a process trying to write to memory as shown in figure 3.2 would be of a great benefit to a student.

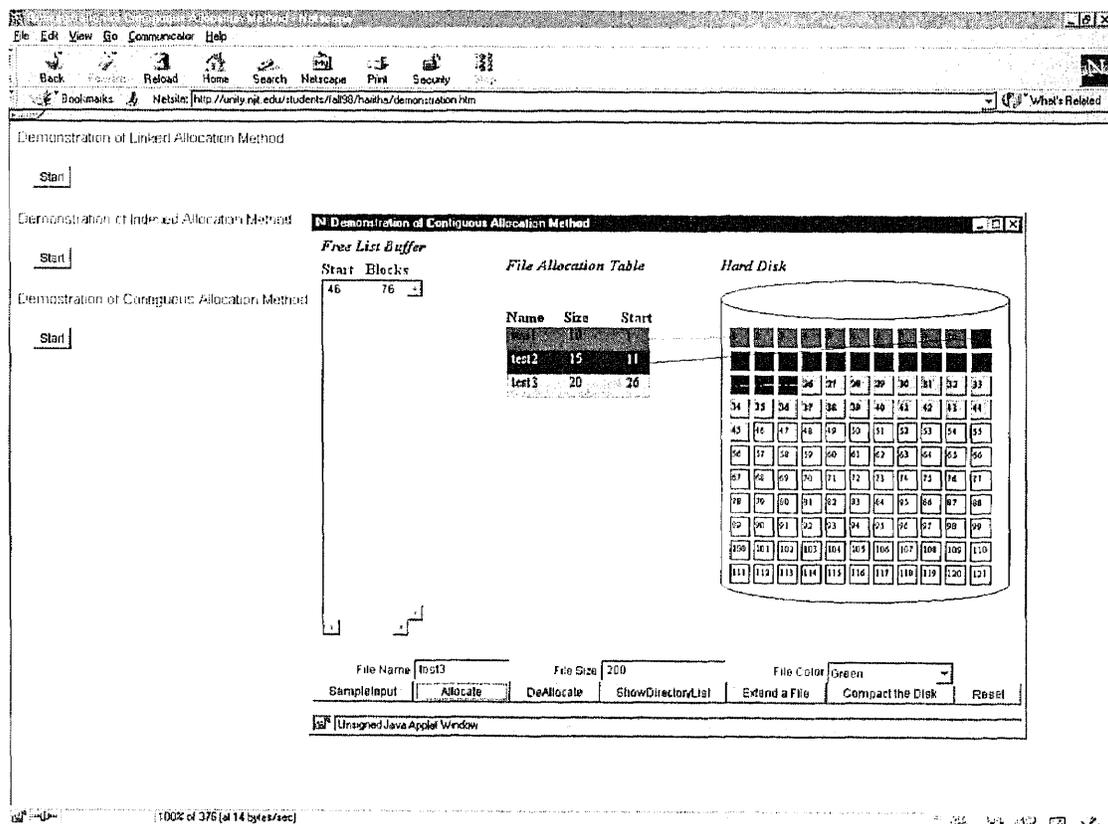


Figure 3.4 Snapshot of an applet interactively explaining contiguous allocation

The interactive quizzes were developed as part of the interactive experience as well.

The quiz shown in figure 3.3 tests the student/user on process management conceptual questions once the s/he has studied the material. The quiz greatly increases the chances of the student retaining the material just studied.

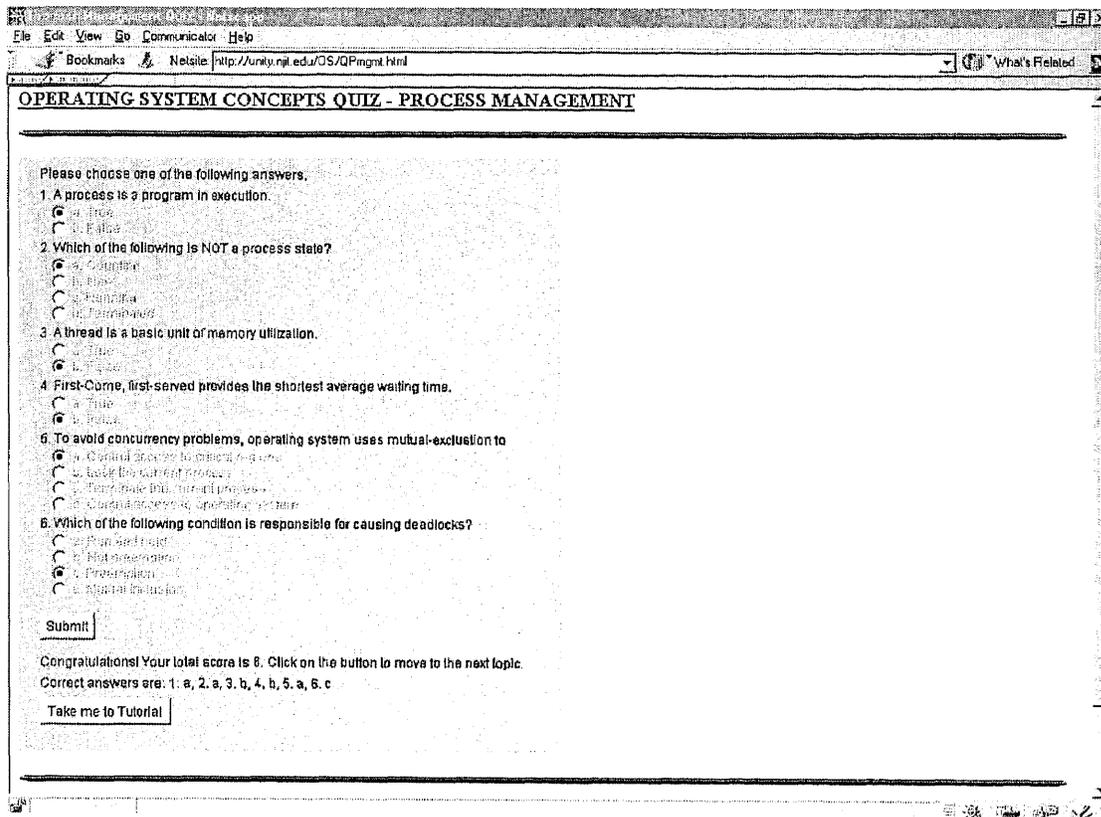


Figure 3.5 Snapshot of an interactive quiz applet

The multimedia quiz is a very interactive quiz created using the Macromedia's Director. It tests the general knowledge of operating system concepts. Based on the score the user is determined to be a beginner, intermediate or an expert and is advised to proceed accordingly. The screenshot in figure 3.4 shows the users inputs during the quiz.

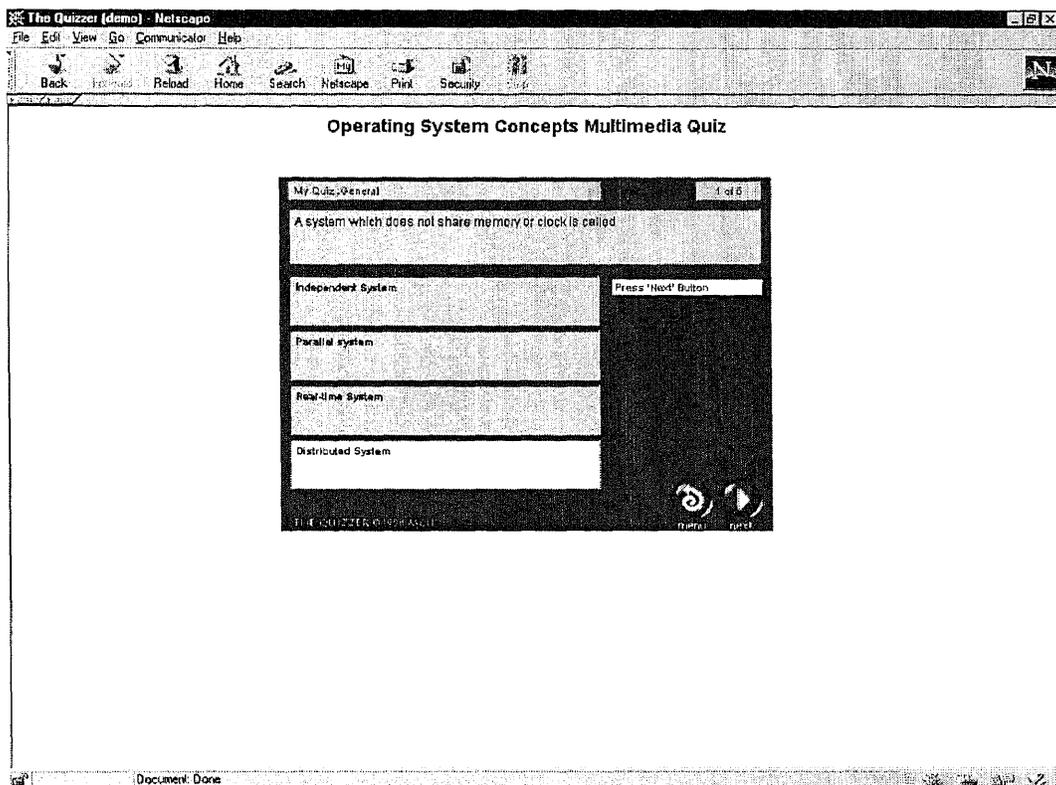


Figure 3.6 Snapshot of an interactive quiz created by Shockwave

CHAPTER 4

USER INTERFACE SPECIFICATIONS

4.1 Introduction

To develop quality software, an intense study of user-friendly interface-design and its issues is of tremendous benefit. During the study, I developed some guidelines that would be used as the “standards” for the current project and future work as discussed with Prof. Kurfess. The guidelines are based on the study of human-computer interaction- HCI (C. Faulkner, 2). It is the study of the relationship existing between human users and the computer systems they use in the performance of their various tasks. It endeavors to provide an understanding of both the human user and the computer system, in an effort to make the interactions between the two easier and more satisfying. To understand users it is necessary to understand the processes, capabilities and predilections that they might bring to the tasks they perform. The will involved an understanding and knowledge of such things as memory, vision, cognition, hearing, touch and motor skills. The software will need to be understood in terms of what it can do for users and how it might best communicate with them. The task of a developer is to design for people, for tasks and for environments. Development a good software system needs to gain its inputs from many other associated areas of study because it has to understand the computer system, the human user and the task the user is performing. The ability to develop a computer system will require an understanding of computer engineering, programming languages, input/output devices and so on. An understanding of the user will require an appreciation of human behavior, of social interaction, of environment, attitudes,

motivation, and so on. An understanding of task requires a means of identifying what is being done and why and in what type of environment. Experts from different fields described in the Figure 4.1.1 can bring wealth of knowledge in design and development process.

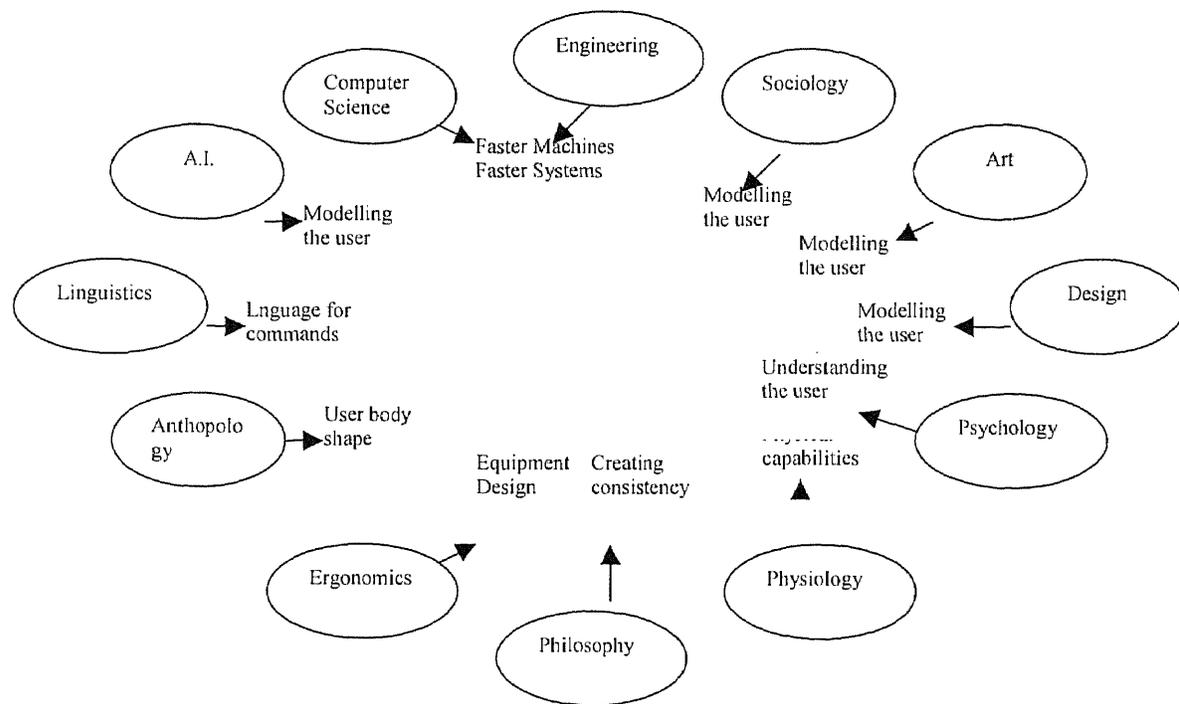


Figure 4.1.1 Various disciplines and their contributions (C. Faulkner, 4)

Information about the world is gathered through the various senses before interpretation can take. These senses can be classified as follows:

- vision
- hearing
- taste
- smell
- touch

4.2 Design Considerations

At the periphery of color vision, the eye is least sensitive to red, green and yellow light and most sensitive to blue light. For this reason, a light color (i.e. gray, beige, sky-blue) makes a good background color, especially where the screen is large. At the front of the eye, where color vision is at its best, the eye is most sensitive to red and yellow and is least sensitive to blue. Small blue objects tend to disappear on the screen, and this especially true where the blue is pale. Furthermore, the eye is not color corrected so that warm colors (reds and yellows) are seen as coming towards the viewer and cold colors are seen as going away from the viewer. This could be used in the development of graphical user interfaces to suggest levels of activity. As always the rule of thumb is to experiment so far as color is concerned and find out which combinations appear to best in testing. Hard and fast rules where human beings are concerned are always dangerous. Because vision is a result of interpretation by the brain as well as information gathering by the eye, it is important to ensure that the layout adopted for the screen is visually pleasing and effective. This means making certain that the screen is not cluttered and that menu choices are arranged in a pleasing way. It is important to stick to the same sort of font, for example, as different types of font mixed together on a screen can lead to a jumbled effect. For example, this paper would be highly unreadable if every so often the font changed to something else and for no apparent reason.

4.3 Principles of Interface Design

Natural

A good interface appears to be natural, it should seem to be an appropriate way of performing the task. The interface should reflect the user's task syntax and semantics: it should be in a natural language for the task involved and should appear to be structured to that task. It should adopt an instructional tone since it has been shown that user become discontent with systems that try to be personal. It should be self-explanatory. The interface will not require the user to do something to the information before it is entered into the system nor do something to the information when it emerges. The system should adapt to the needs of the user and not expect the user to adapt to its needs.

Consistency

The interface should reinforce the user's expectations from any previous interaction with that system or indeed with similar systems. It should be consistent in its requirements for input and have consistent mechanisms for the user to make any demands on the system. The language of the system messages and the system prompts should always be the same and should be in the same place where it is possible. The output screen format should be consistent throughout the application. Any changes to the format should be done for good reasons and should be obvious. There should be consistent format for menus, messages and so on.

Relevance

The interface should not ask for redundant material. It should require the minimum of user input and should provide the minimum of system output necessary for the completion of the user's task. The user does not have time to read a lot of unnecessary material so on-screen information should be short and relevant but at the same time it must make sense. There should be minimum keyset/keystroke effort. This will maximize user performance and prevent unnecessary errors from occurring by reducing the number of potential key presses. This especially important where the user is not a proficient typist.

Supportiveness

The interface should provide adequate information to allow the user to operate and to perform the task. This means that the system should provide sufficient specific help in order to help users formulate input requests. There should be adequate feedback providing information to help the user continue with the task. For example, the sort of questions the user may well want answered are:

- Where am I?
- How did I get here?
- What is happening?
- Where I go next?
- How do I get there?
- What can I do?

Flexibility

The interface should accommodate differences in user requirements and also in user preferences and level of performance. This will mean that the system needs to preserve consistency for the individual user, while at the same time recognizing that so far as different users are concerned there may well be a need to tolerate a wide range of input syntax and semantics. The interface will also need to provide a variety of support levels and should also allow personalized output formats. In other words, the system should gear itself to the range of needs required by the different user types and their tasks.

4.4 User Classification

For the sake of convenience, it is possible to categorize the user into user types. For example, so far as levels of system acquisition are concerned the following three categories of user should be identified:

- novice
- knowledgeable intermittent
- expert/frequent users

Of course, in extreme cases the nature of the user population might not be known and it will therefore be assumed that it might consist of any level of ability. In these cases artificial boundaries can be drawn as would be done if the design was aimed at the general public.

Novice Users

For the novice user of software, processing is slow and laborious because of the limitations of working memory. Systems to be used by novices require more feedback and more opportunities for closure. The novice needs to feel that progress towards the goal is being made and is not left for long periods of time wondering if what has been done so far is right as this might well cause anxiety. The undergraduate class of operating system comprises mostly of Novice users.

Guidelines for novice users

- 1 All initiatives should come from the computer/software. The novice user will not know what can be done. Novices may or may not have a good understanding of the task domain but even so, the system needs to provide as much help as possible.
- 2 Required input should be brief. Input should be brief because the shorter it is the more likely it is to be remembered and the less likelihood that there is of an error occurring when the user keys in the response.
- 3 Input procedures should be consistent with user expectation. The user should not be faced with anything that seems inconsistent or unlikely.
- 4 No special training should be necessary. All of the information needed to operate the system should be provided by the system itself.
- 5 All system messages should be clear and unequivocal. The user should not be left puzzling over the meaning of a particular message.

- 6 User decisions should be made from a small set of options. Novice users need to make choices from a small number of options available since it makes them feel that the system is limited in size and it avoids feelings of being overwhelmed by the enormity of the system.
- 7 Users should control the pace of interaction.
- 8 User decision making should be in response to a specific request for action. The user should not be expected to guess that it is now time to do something. Decision making needs to be prompted and the system should take the necessary initiative.
- 9 Help should be always available.
- 10 There should be sufficient feedback. Novice users like to know if they are on the right track. There should be opportunities for closure to occur so that the novice feels comfortable about how things have progressed so far.

Knowledgeable Intermittent Users

Knowledgeable intermittent users are able to maintain semantic knowledge both the task they want to perform and of the computer concepts involved. However, because their use of the system is intermittent they need consistent structures, good help facilities, and good documentation. Documentation and help systems are particularly important to the intermittent user as they will know what they want to do and possible the sort of process involved in carrying out the particular task but may need prompting on the details of how this will be done.

Expert Users

Expert users will be well versed in both the semantic and syntactic aspects of the computer system. Their response time will be rapid and they will require brief feedback because they are accustomed to the performance of the system. Experts organize their knowledge according to a higher conceptual structure. They are able to recall larger amounts of information than novices. They will typically look for abbreviated command sequences and will also want accelerators to take them through the dialogue sequences as rapidly as possible. In fact, the requirements of an expert user are almost in contradiction of the requirements of a novice user.

4.5 Design Process

In order to design a system, it is necessary first to understand what is that the system should be doing. The process of understanding the job to be done by the system is provided by task analysis.

Task Analysis

The process of task analysis should produce a clear understanding of what it is that the system must do. The design process should begin with an analysis of what needs to be done. In order to accomplish this, the user's task will need to be examined in some detail. The following questions will have to be answered for each major task carried out:

- What does the performer of task do?
- What information is used for each task?
- What affects task performance?
- What causes error in task performance?
- What skills are required for each task?
- How are these skills acquired?
- How are these skills maintained?

Procedure for Task Analysis

Each task is a hierarchy of task and subtask. In other words tasks can be broken down to their simplest level of task. For each task define:

The inputs to the task

- What information is needed?
- What are the characteristics of the information sources?
- What is the availability of information?
- What possible errors might occur?
- Who or what initiates the task?

The outputs from the task

- What is the performance criteria?
- What happens to the output?
- How does the task performer get feedback about task performance?

The transformations

- What is the nature of the decision making?
- What strategies exist for the decision making?
- What skills are needed?
- What interruptions are likely to occur and when?

The task composition of the particular job

- How often is task done and when?
- Does the task depend on any other task?
- What is normal work load?

It helps to examine similar systems in designing process.

Collection of data about the system

Any of the following methods can be used to collect appropriate information about either existing or projected systems:

- interview with users
- observation
- activity sampling
- activity logging

It is important that the design team gets the confidence of the end users and involves them in the design process. The tasks are sampled or logged or both. Any of the above mentioned evaluation methods could be used as a method of first gathering information about the system.

4.6 Evaluation and Testing of the System

Guidelines

Evaluation is the parallel to systems testing in software engineering. It is the process by which the interface is tested against the needs and practices of the user. Just as in software engineering it is necessary to have a clear set of goals for test testing of software so it is in testing a system with the user. Therefore, it is important to decide what aspect of, when and how the user interface will be tested and by whom the testing will be done.

Problem Areas in Evaluation

The testing of the system, like the design process, is faced with problems and difficulties. Following is a list of relatively common problem areas in testing:

Common sense

It is tempting to assume that because something seems obvious it must be correct. It is best to try not to avoid making assumptions about anything.

Testing it on yourself

Testing a system on yourself solely is another dangerous mistake since the designer of a system can hardly be expected to be typical. The problem is that the designer has skills that the average person/user does not have.

Consulting the real end user

For a real understanding of what actually occurs you have to speak to the people performing the tasks. Task performers know what they do and design teams must speak to them.

It can be evaluated later...

Design decisions can be difficult to reverse if evaluation appears late. It should be seen as an on-going process: design-evaluate-redesign.

CHAPTER 5

CONCLUSIONS

Although interactive multimedia offers great promise, design is not easy. Indeed, a developer faces all the traditional aesthetic and functional design challenges faced by those who work in single text, image, or sound media plus complex new ones. These challenges include the following:

- **Time Design:** How does one organize events that unfold in time? While this question is not new to those who work in time based media such as cinema, video, music, theater, or dance, it is new to all those entering multimedia design from other fields.
- **Coordination of Media:** How does one ensure that simultaneous text, image, and sound work together to achieve functional or aesthetic intentions? Historically, media have worked extensively with the problems of coordination - for example, cinema's attention to sound and image. The need, however, to coordinate them on a single electronic display is unprecedented. Also, as the sophistication of computers and their displays increase, the possibilities of multiplying the media elements included will expand into uncharted areas (for example, the simultaneous display of several video and sound elements or the creation of virtual reality immersion environments.)
- **Interactivity:** How does one design events so that the process by which a user navigates and makes choices is engaging and/or effective? The designer faces unprecedented challenges of addressing the psychology of users and creation of choice/user action structures.

The World Wide Web is the ultimate undoing of traditional education. Suddenly, your desktop is all the libraries, all the newspapers, all the universities, and all the other desktops in the world. If you have a computer, or access to one, you can find out virtually anything you want to know. And using it as an educational tool should be common sense. When incorporate with multimedia, the final result is a powerful educational tool. But putting a book on a CD-ROM doesn't make it multimedia. Turning electronic pages isn't interactive. The total multimedia experience is where the user gets involved so deeply in the experience that the user loses the sight of the surroundings and track of time. The gaming industry fully explores the potential of the medium. Myst, Doom, and SymCity engage their users precisely because the user must learn the rules of the environment in order to advance. The goal should be to redirect this dynamic environment that so consumes gaming enthusiasts. We should use it to teach students in a similar way. In pursuit of this goal, we abide by some fundamental concepts:

- No one teaches a child how to learn. In real life, learning happens by trial and error. Ideally, a parent or mentor provides guidelines and encouragement.

Learning should harness the power of this trial-and-error process, while providing the context, guidance and feedback of a parent or mentor.

- Education should provide room for experimentation with consequences that mimic real life. You don't get a score in life; you get a paycheck. A student should be challenged with a real-life problem scenario.

- A multimedia education program should function as a learning guide while also serving as a resource for just-in-time reference. People need answers to questions as they occur to them, not when an instructor is available.
- In creating content, two problems emerge. Experts don't remember where they began, and beginners don't know where they need to end up. For a creation of an effective interactive experience, developer needs to work extensively with both beginner and expert, a process largely ignored.
- The learning experience should be fun. It should be engaging. It should include sound, graphics and appropriate animation. It should stimulate interest and questions, and then provide the resources to find the answers.
- Learning can be as engaging as Myst or Doom or SymCity. That multi-sensory experience combined with dynamic content can teach someone a concept that will never be forgotten.

While developing an interactive software is difficult, the task can be made easier by following the guidelines described in chapter 4. Some of the suggestions in addition to the guidelines are:

- A detailed study needs to be taken of the students who would be the end users.
- Several surveys should be handed out in the classrooms asking students for their input.
- The input of professors/teachers should be used extensive throughout the development.
- The opinions of the recent graduates should be very valuable in determining beneficial and detrimental tools and teaching methods.

APPENDIX A

USER MANUAL

Although the web-site is quite interactive, there is a need for a user manual for applets animating the operating system concepts. The help is broken down by each applet.

Main Screen

The main page represents following four choices as well as other necessary steps.

- Click on Take the General Quiz link to take the quiz which will test you on topics ranging from introductory general computer system structure to more advanced topics such as distributed networks. This quiz will also determine your knowledge level.
- Click on Take me to Tutorial link to visit the tutorial containing definitions, colorful figures, and animated applets depicting various operating system concepts.
- Click on Explore more resource link to explore further. This useful to someone who is at the expert level or one who has covered all the topics on the web-site.
- Click on Sign the Guestbook link to sign the guestbook to reflect your opinion on your interactive educational experience or to share ideas with others or to give us suggestions to improve the site.

The "Download" button leads you to another page contains links to the websites which let you download necessary multimedia tools such as Shockwave or java-enabled browsers necessary to take quizzes and view animated applets. The "Email" button allows you to send an email to the administrator for any questions/ comments.

Interactive Quizzes

The quiz will test your knowledge on the operating system concepts such as memory management, process management, protection and security, and networks or distributed systems. Based on your score, you will be guided to the appropriate topic.

The steps are following:

1. Select a correct answer from given choices
2. When finished, click on the *Submit* button to find out your score.
3. Press *Take me to Tutorial* button to move to the appropriate section.

Interactive Process Scheduling Applet

The user can choose any one of the following different algorithms from CPU

Algorithm Choice box:

1. Round Robin Scheduling(RRS)
2. Shortest Job First Scheduling(SJF)
3. Priority Scheduling(PS)
4. First Come First Serve Scheduling(FCFS)

The user then enters number of processes and press “OK” button. If the user doesn’t enter appropriate data and press “OK” button, the error message pops up. If the user enters all the appropriate data in proper format and presses “OK” button, the user will see a table under Process Input box that is created dynamically depending on the number of processes the user entered. And then the user can enter the arrival time and burst time of the processes and so on. When the user presses “Result” button, the user will see Overall Result which is a table of description of each process’

performance including Process ID, arrival time, burst time, waiting time, process time, turnaround time and context switch. Right below the table, you will see the average waiting time of the algorithm. In Time Frame Result Box, you will see which process was using CPU at each time. This table is also used to draw the Gantt Chart below it and to match up the user's guess when the user decide to see the result step by step and try the next step.

Multi-level Feedback Queue Scheduling Applet

This applet demonstrates multi-level feedback for the types of process scheduling as follows:

1. Round Robin Scheduling(RRS)
2. Shortest Job First Scheduling(SJF)
3. Priority Scheduling(PS)
4. First Come First Serve Scheduling(FCFS)

The user can choose any of the above scheduling methods for each process. The input fields required are process id, arrival time, execution time, priority, and the type of scheduling. The process id is an integer is used to distinguish between multiple processes. Arrival time is also an integer number indicating time the process arrived in the process queue. Execution time is an integer assigning the total CPU time needed to complete the process. Priority number is used only for the Priority Scheduling assigning a priority to each process. A low number indicates a higher priority. Type of scheduling allows you to choose from one of the types described above. Use the "Add", "Delete", and "Clear" buttons to add/modify/delete the

processes in the queue with given criteria. Once the processes are in the queue, they are ready for simulation. Click on "Run" button to start the animation. Multi-level-feedback-queue shows the colorful graphical animation of process runs while the statistical view shows the numerical representation of simulation.

Unix Process Manager Applet

The applet simulates total five processes with selected parameters executed by the unix kernel. When the initial screen comes up, select "Set" from the Process Parameters menu. The parameters to be set are start time of process (from 1-10) and the life span of process (from 1-10). Once the parameters are selected, select "Run Simulation" option from the Simulator menu. The colorful animation represents different processes by yellow, blue, red, white, and black colors. The process parameters shown on the screen are priorities, which are randomly assigned. It also shows the start time and time left in process run. The different process states include born, ready, running, and zombie.

Process Synchronization Simulation - Sleeping Barber Applet

A classic synchronization problem which addresses race conditions & deadlocks. In order to start the simulation, you need to click the "Barber" button to bring the barber in the shop. To bring customer in, you need to click on the "Customer" button. You can bring in as many customers as you want by clicking on the button

Now you need to see the barber's chair is free or not, if its free then the customer will automatically get a chance to get a hair cut, otherwise the customer will have to wait

in the waiting chair. When the customer gets a haircut, customer will leave the shop. If there is no customer, then the barber will click on sleeping chair and falls asleep. The barber services one customer at a time and once the barber is done with the current customer, barber will wake up the next waiting customer.

Deadlock Simulation - Dining Philosophers Applet

"Restart" button will restart the animation, incase there is a deadlock or if the user has pressed freeze. "Freeze" button stops the animation and displays the current state of the philosophers and forks at the table. "Thinking/Eating time" slider controls the time a philosopher takes to think and eat. The thinking time is twice the eating time. User needs to decrease the slider to a minimum to observe deadlock in the deadlock version of the program. System State displays the state of the system. You can one of the two algorithm, one of which displays a simulation which will deadlock if the timer is reduced sufficiently, and the other one which will not deadlock even if the slider is moved lower or higher.

Deadlock Detection System Applet

A simple yet quite effective applet animating deadlock detection allowing you to simulate and detect a deadlock. This applet allows you to select a process and allocate a resource such as a printer, disk, etc. Processes have a choice of either requesting a resource or holding a resource for the purpose of the simulation. Once the process are assigned the resource, click "Start" to simulate process. The output shows the message whether the deadlock was present or not.

Paging Simulation Applet

This colorful applet simulates the paging scheme used by the operating system. First, to start a process insert number of pages required during the execution in the “Pages” fill-in field. Assuming that a long process require large number of pages while a small process requiring less computing power requires fewer number of pages. Using the assumption, once the pages are assigned to a process, click on the “Start New Process”. Once the process is completed and the assigned pages are used, select “Calculate Address” button, to get the logical address as well as the physical (or actual) addressing used by the OS. This information is also described pictorially in the bottom drawing on the simulation. You have an option to select either Decimal, Hexadecimal or Octal system to calculate the address space.

Relocation Simulation Applet

Relocation applet simulates the run-time mapping done by the memory-management unit (MMU) to map the virtual addresses to physical addresses. First, you need to select either Decimal, Hexadecimal or Octal system in address calculations. Then, input a logical address in the fill-in field “Logical Address”. The field next to it is the address of the “Base Register”. When the values for these two fields are entered, click on the “Calculate Address” button to start the mapping. The end-result shows the physical address is calculated by adding the logical address to the initial address. The physical address is shown in the lower right corner in the fill-in field “Physical Address”.

Linked Allocation Simulation Applet

With linked allocation, each file is a linked list of disk blocks, the disk blocks can be scattered anywhere on the disk. The directory contains a pointer to the first and last blocks of the file. When the applet starts, you can first understand the process by simply pressing the “SampleInput” button simulates the whole process without any user input. The applet has three fill-in text boxes. First is the file name where the name of the file to be allocated is entered. Second field is the size of the files. And the final field is the color schema for the block mapping on the “HardDisk” graph to distinguish between different files. Once the fields are selected, then you need to click on the “Allocate” button to start the file creation process. The simulation shows the File Allocation Table (FAT) in the middle of the graphics. The right side is the layout of the file in the Hard Disk. And finally the Free List Buffer shows the available blocks and their starting points. The “DeAllocate” process allows a file to be deleted from the Hard Disk and thus simulating the file deletion. The button “ShowDirectoryList” shows the files created in the current directory. It is like “dir” and “ls” commands in the MS-DOS and UNIX operating system, respectively. “Extend a file” simulates the file update process where a file can size could later be increased or decreased once it has been created. “Reset” button allows you to clear all inputs and start over.

Indexed Allocation Simulation Applet

Linked allocation solves the external-fragmentation and size-declaration problems. Here each file has its own index block, which is an array of disk-block addresses. The screen is same as the linked allocation simulation described above.

Contiguous Allocation Simulation Applet

Contiguous allocation method requires each file to occupy a set of contiguous blocks on the disk. The addresses are ordered in linear manner. The screen is same as the linked allocation simulation described above. However, there is one additional feature of this simulation which is the “Compact the disk” button. This feature allows one to remove any unused space left on the disk. It requires other blocks to be moved also. In the real world, this process takes significant time and is usually run when the disk is very fragmented.

APPENDIX B

JAVA APPLLET CLASS DEFINITIONS

Interactive Quiz – Process Management

The following class definition is used for the creation of the process management quiz applet embedded in the browser. Similar definitions are used for other interactive quizzes.

```
public class Pmgmt_quiz extends Applet implements ActionListener {
    String msg1 = "";
    String msg2 = "";

    int correct=0;

    CheckboxGroup g1 = new CheckboxGroup();
    CheckboxGroup g2 = new CheckboxGroup();
    CheckboxGroup g3 = new CheckboxGroup();
    CheckboxGroup g4 = new CheckboxGroup();
    CheckboxGroup g5 = new CheckboxGroup();
    CheckboxGroup g6 = new CheckboxGroup();
}
```

Multi-Level Queue Process Scheduling Applet

There are mainly 4 classes used to store data, namely class MLQueue , class process, class processData, and class OutputData.

```
class process // Stores process information {
    public int id;
    public int arrTime;
    public int exeTime;
    public int priority;
    public int Qnum;

    public int burTime;
    public int waitingTime;
    public int finishTime;
    public int turnaroundTime;
};
```

```

class MLQueue {
    public int front;
    public int rear;
    public int count;
};

public class ProcessData {
    int Id;
    int ArrivalTime;
    int ServiceTime;
    int priority;
    int queueNum;
}

public class OutputData {
    int Id;
    int queueNum;
}

```

Sleeping Barber Applet

The following class definition is used to create the BarberShop applet animating the sleeping barber problem.

```

public class BarberShop extends Applet implements Runnable {
    Panel Buttonpanel;
    Button Barber, Customer, Stop, Race, Deadlock, Resume;
    Messagearea message= new Messagearea();
    static boolean barberavailable, barbersleep;
    static boolean customertoserve[] = new boolean[11];
    circle barber;
    circle customers[]=new circle[11];
    static ThreadGroup customeralive;
    static Thread customeralives;
    //static int index=0;
    Graphics g;
}

```

Disk Block Allocation Applet

The `clBottomPanel` class is used in the Disk Block Allocation applet to convey contiguous, linked, and indexed allocation concepts.

```
public class clBottomPanel extends Panel {
    GridBagLayout gridbag;
    GridBagConstraints constraints;

    //All objects
    clTopPanel TPanel;
    clDialogBox d1;
    clTwoPanels BaseFrm;
    clFreeListBuffer FLB;
    clFLBPanel FLBPanel;

    //This vector is used to temporarily store the vFileDetails vector during
    compacting.
    static Vector vTempFD = new Vector();

    //All the components
    Label lblFileName;
    TextField txtFileName;
    TextField txtFileSize;
    Label lblFileSize;
    Label lblFileColor;
    Choice chFileColor;
    Button btAllocate;
    Button btDeAllocate;
    Button btShowDirectoryList;
    Button btSample;
    Button btReset;
    Button btExtent;
    Button btCompact;

    //This vector is used for File Details
    static Vector vFileDetails = new Vector();

    //this contains the list of allocated files, their colors and Xstart and Ystart
    positions
    //of rects which are in queue to be allocated.
    //static Vector vPendingAllocFiles = new Vector();

    //contains free list buffer
    static Vector vFreeListBuffer = new Vector();
```

```
//This acts as index in vFileDetails
static int iFileIxInclFileDetails;

static int iNumBlocks;

//Extent size in Number of Blocks
//final int iBlksInExtent = 4;

//length of the string that is displayed in FLB
static int iLenstrDisplayFLB = clFLBPanel.iLenStrFLB;

static boolean bSampleInputFlag = false;

//BottomPanel constructor
clBottomPanel(clTopPanel topPanel,clFLBPanel flbPanel) {
    //String strFLB;
    TPanel = topPanel;
    TPanel.setBottomPanel(this);
    FLBPanel = flbPanel;
}
}
```

REFERENCES

1. C. Faulkner, *The Essence of Human-Computer Interaction*, Prentice Hall, New York, 1998.
2. C. S. Horstmann and Cornell, Gary, *Core Java*, The Sunsoft series, Mountain View, California, 1997.
3. A. M. Lister and R. D. Eager, *Fundamentals of Operating Systems*, Fifth Edition, Springer-Verlam, New York, 1993.
4. J. Koenemann and N.J. Belkin, *Conference Proceedings Chicago '96- A Case for Interaction: A Study of Interactive Information Retrieval Behavior and Effectiveness*, Chicago, Illinois, 1996.
5. C. Pearce, *The Interactive Book*, Macmillan Technical Publishing, Indianapolis, Indianapolis, 1997.
6. A. Silberschwatz and P. B. Galvin, *Operating System Concepts*, Addison-Wesley Publishing Co, Boston, Massachusetts, 1994.
7. A. Tanenbaum, *Operating Systems- Design and Implementation*, Prentice-Hall, Englewood Cliffs, NJ, 1997.