Many instructors believe that implementation or research projects are crucial to the clear understanding of operating system concepts. Without projects, it may be difficult for students to grasp some of the basic OS abstractions and interactions among components; a good example of a concept that many students find difficult to master is that of semaphores. Projects reinforce the concepts introduced in this book, give the student a greater appreciation of how the different pieces of an OS fit together, and can motivate students and give them confidence that they are capable of not only understanding but implementing the details of an OS.

In this text, I have tried to present the concepts of OS internals as clearly as possible and have provided numerous homework problems to reinforce those concepts. However, many instructors will wish to supplement this material with projects. This appendix provides some guidance in that regard and describes support material available at the instructor's Web site. Further details are provided in Appendices D and E.

C.1 PROJECTS FOR TEACHING OPERATING SYSTEMS

The instructor can choose from the following approaches.

- **Operating Systems Projects (OSP):** OSP is both an implementation of a modern operating system and a flexible environment for generating implementation projects appropriate for an introductory course in OS design. OSP is accompanied by a number of project assignments.

- **Ben-Ari Concurrent Interpreter (BACI):** BACI simulates concurrent process execution and supports binary and counting semaphores and monitors. BACI is accompanied by a number of project assignments to be used to reinforce concurrency concepts.

- **Nachos:** As with OSP, Nachos is an environment for generating implementation projects to reinforce concepts and is also accompanied by a number of project assignments.
• **Research projects:** The instructor's Web site provides a series of research assignments that assign the student to research a particular topic on the Internet and write a report.

• **Programming projects:** The instructor's Web site provides a set of small programming projects that can be assigned to reinforce concepts in the book; any language can be used. The projects cover a wide range of topics dealt with in this book.

• **Reading/report assignments:** The instructor's Web site includes a list of important papers, one or more for each chapter, that can be assigned to students to produce a short report analyzing the paper.

This appendix provides a brief discussion of these topics. Appendix D provides a more detailed introduction to OSP, with information about how to obtain the system and the programming assignments. Appendix E provides the same type of information for BACI. Nachos is well documented at its Web site and is described briefly in the next section.

### C.2 NACHOS

**Nachos Overview**

Nachos is an instructional operating system that runs as a UNIX process, to provide students with a reproducible debugging environment, and that simulates an operating system and its underlying hardware [CHRI93]. The goal of Nachos is to provide a project environment that is realistic enough to show how real operating systems work yet simple enough that students can understand and modify it in significant ways.

A free distribution package is available via the Web that includes

- An overview paper.
- Simple baseline code for a working operating system.
- A simulator for a generic personal computer/workstation.
Sample assignments: The assignments illustrate and explore all areas of modern operating systems, including threads and concurrency, multiprogramming, system calls, virtual memory, software-loaded TLBs, file systems, network protocols, remote procedure calls, and distributed systems.

A C++ primer (Nachos is written in an easy-to-learn subset of C++, and the primer helps teach C programmers this subset).

Nachos has been used at hundreds of universities around the world and has been ported to numerous systems, including Linux, FreeBSD, NetBSD, DEC MIPS, DEC Alpha, Sun Solaris, SGI IRIX, HP-UX, IBM AIX, MS-DOS, and Apple Macintosh. Future plans include a port to Stanford's SimOS, a complete machine simulation of an SGI workstation.

Nachos is freely available from its Web site (there is a link to their Web site from WilliamStallings.com/OS/OS5e.html); a solution set is available to instructors by e-mail from nachos@cs.berkeley.edu. In addition, there is a mailing list for instructors and a newsgroup (alt.os.nachos).

Choosing among Nachos, OSP, and BACI

If the instructor is willing to take the time to port one of these three simulators to the local environment available to the students, then the choice among these three will depend on the instructor's objectives and personal opinion. If the focus of the projects is to be on concurrency, then BACI is the clear choice. BACI provides an excellent environment for studying the intricacies and subtleties of semaphores, monitors, and concurrent programming.

If, instead, the instructor wishes to have students explore a variety of OS mechanisms, including concurrent programming, address spaces and scheduling, virtual memory, file systems, networking, and so on, then either Nachos or OSP may be used.

I have included an appendix on OSP because I feel it is one of the best vehicles available for supporting OS projects. OSP is in use at over 100 sites and provides a great deal of support.
and documentation. One potential drawback is that although the system, sample assignments, and mailing-list support are free, there is a small user's manual that students are expected to purchase. However, this must be balanced against the strengths of this environment. Nachos is similarly widely used and provides support, documentation, and suggested assignments. The instructor is urged to study Appendix D and, if interested, compare it to the Nachos overview paper and other documentation available at the Nachos Web site.

C.3 RESEARCH PROJECTS

An effective way of reinforcing basic concepts from the course and for teaching students research skills is to assign a research project. Such a project could involve a literature search as well as a Web search of vendor products, research lab activities, and standardization efforts. Projects could be assigned to teams or, for smaller projects, to individuals. In any case, it is best to require some sort of project proposal early in the term, giving the instructor time to evaluate the proposal for appropriate topic and appropriate level of effort. Student handouts for research projects should include:

- A format for the proposal
- A format for the final report
- A schedule with intermediate and final deadlines
- A list of possible project topics

The students can select one of the listed topics or devise their own comparable project. The instructor's Web site includes a suggested format for the proposal and final report as well as a list of possible research topics developed by Professor Tan N. Nguyen of George Mason University.

C.4 PROGRAMMING PROJECTS
An alternative to the development of portions of an OS, using OSP or Nachos, or concentrating on concurrency, using BACI, is to assign a number of programming projects that require no infrastructure. There are several advantages of programming projects versus the use of a support framework such as OSP or BACI:

1. The instructor can choose from a wide variety of OS-related concepts to assign projects, not just those that would fit in the support framework.
2. The projects can be programmed by the students on any available computer and in any appropriate language: they are platform- and language-independent.
3. The instructor need not download, install, and configure the infrastructure.

There is also flexibility in the size of projects. Larger projects give students more a sense of achievement, but students with less ability or fewer organizational skills can be left behind. Larger projects usually elicit more overall effort from the best students. Smaller projects can have a higher concepts-to-code ratio, and because more of them can be assigned, the opportunity exists to address a variety of different areas. On balance, the advantage seems to lie with smaller projects. Accordingly, the instructor's Web site contains a series of small projects, each intended to be completed in a week or so, which can be very satisfying to both student and teacher. These projects were developed by Stephen Taylor at Worcester Polytechnic Institute, who has used and refined the projects in the course of teaching operating systems a dozen times.

In addition, two more substantial programming projects are outlined in the text, after Chapters 3 and 9. A more detailed, step-by-step set of instructions for these two projects is provided at the instructor's Web site.

C.5 READING/REPORT ASSIGNMENTS
Another excellent way to reinforce concepts from the course and to give students research experience is to assign papers from the literature to be read and analyzed. The instructor's Web site includes a suggested list of papers to be assigned, organized by chapter. All of the papers are readily available either via the Internet or in any good college technical library. The instructor's Web site also includes a suggested assignment wording.