Operating Systems with Linux: Foundations and High-Performance Computing (HPC)

SGX3 award # 2231406

Revised Course Description

Current Course Description

This course introduces students to the principles and concepts of modern operating systems design, discusses major issues of importance in the design, and shows how operating systems have implemented the design ideas. Topics include process management, CPU scheduling, memory, and storage management. In addition, this course will specifically introduce students to the Linux operating system, its commands and programming features.

Proposed Course Description

This revised course introduces students to the principles and concepts of modern operating system design, with a focus on the Linux operating system and its application in High-Performance Computing (HPC) environments. The course covers key topics such as process management, CPU scheduling, memory and storage management, while also providing an introduction to parallel computing and distributed systems. Students will learn essential Linux commands, programming features, and gain practical experience with tools and environments used in HPC. Through hands-on lab exercises and projects, students will explore how to work effectively in an HPC environment.

Implementation Schedule

The course will be offered in the Fall Semester, from 2025.

Weeks 1-2: Overview of Operating Systems (OS)
Introduction, OS structures, Linux basic commands, command-line interface (shell), Linux in HPC.

Weeks 3-4: Process Management
Processes, threads, synchronization, algorithms and deadlocks, process management system calls in Linux, process management in HPC environments.

Week 5-6: CPU Scheduling
Scheduling algorithms, Linux CPU scheduling, introduction to HPC job scheduling (e.g., Slurm), resource management in HPC.

Week 7-8: Memory Management
Main memory, virtual memory, page replacement algorithms, memory management system calls in Linux, memory tuning in HPC environments

Week 9-10: Storage and I/O Management
File systems, storage structures, file-system management, Linux I/O system calls, HPC file systems and I/O performance

Week 11-14: Parallel Computing & Performance, Advanced Topics
Introduction to parallel computing, performance benchmarking in HPC environments. Protection and security, virtual machines, distributed systems, introduction to cloud-based HPC environments.

Sample HPC/Gateways Exercise

1. Exploring Linux Commands for HPC
Familiarize students with essential Linux commands and file system navigation used in HPC environments. Use commands like *pwd, ls, cp, mv, rm*, and *mkdir* to navigate and manage files.

2. Process Management in Linux for HPC
Explore process management, system calls, and process control in Linux. List running processes using *ps aux* and *htop*. Create and manage processes using system calls like *fork(), exec(), and wait()* in a simple Bash script.

3. CPU Scheduling and Resource Management in HPC
Explore CPU scheduling and process prioritization, emphasizing HPC resource management.

4. Memory Management and Virtual Memory in Linux
Investigate memory usage and virtual memory management in Linux, crucial for HPC performance tuning. Write a Bash script to check memory usage periodically and alert when it exceeds a specified threshold.

5. File Systems and I/O Management in HPC
Explore file-system and I/O management in Linux, with an emphasis on HPC environments. Manage disk space with *df -h* and experiment with mounting and unmounting storage devices.

6. Introduction to Parallel Computing with MPI
Implement parallel computing tasks using MPI to demonstrate efficient computation in HPC environments.

Resource Needs/List

1. Access to HPC Platform with Ubuntu/Mint distribution for student account creation and experimentation.

2. Computer lab or personal computers with internet access.

3. Text Editors and IDEs (Vim, Nano, or Visual Studio Code)

4. Recommended online resources on Linux man pages or documentation for system calls and commands.

Resources / Science Gateways

- ACCESS-CI
- TACC
- Science Gateways Community Institute (SGCI)
- ORNL Portal

Gateway Community Mentor Syllabus Suggestions

The mentor recommended that the lab exercises be closely aligned with teaching students how to work effectively in an HPC environment. This approach involves progressively introducing core operating system concepts, followed by specialized HPC topics, to provide both theoretical and practical insights. The suggested enhancements aim to strengthen the connection between standard OS topics and the practicalities of working in HPC.

Additionally, the mentor emphasized the importance of integrating foundational OS concepts with relevant HPC topics in the course schedule, ensuring that students build a solid understanding before advancing to more complex HPC applications.

Possible Expansions

To enhance the curriculum and provide students with a comprehensive understanding of High-Performance Computing (HPC) concepts, we propose the development of a dedicated undergraduate course titled "**Distributed Computing**." This course would cover essential topics that extend beyond the operating systems curriculum, focusing on the principles and techniques used in distributed systems.

Course Highlights:

- **Introduction to Distributed Systems**: Students will learn the fundamental concepts, architecture, and design principles that underpin distributed computing environments.
- **Communication Models**: Explore various communication models used in distributed systems, including message passing, remote procedure calls, and data streams.
- **Synchronization and Coordination**: Study synchronization techniques and coordination mechanisms to manage concurrent processes and ensure consistency in distributed environments.
- **Fault Tolerance and Reliability**: Understand strategies for building reliable distributed systems that can recover from failures and maintain operational continuity.
- **Real-World Applications**: Analyze case studies of distributed systems in practice, such as cloud computing, peer-to-peer networks, and large-scale data processing frameworks.

By offering this course as a special topic within the Computer Science department, students will gain in-depth knowledge of distributed computing, equipping them with the skills needed to tackle complex challenges in modern computing environments. This expansion will not only benefit students' understanding of HPC but also expose them to diverse areas of technology, allowing them to explore career paths beyond the traditional software engineering roles and better prepare them for the rapidly evolving technology landscape.

Use Cases

- High-Throughput Computing
- Scientific Simulations
- Parallel Algorithm Development

Authors

Olamide T. Tawose, PhD.
Department of Computer Science
Lincoln University of Pennsylvania
otawose@lincoln.edu

HPC/Gateways Mentor
Hector Corzo, PhD.
National Center for Computational Sciences, Oak Ridge National Laboratory
hernandezchf@ornl.gov

Special Thanks

- Dr. Linda Bailey Hayden, SGCI/SGX3 Director of Workforce Development.
- Je'aime Powell, Texas Advanced Computing Center.
- All Faculty Mentors during the SGX3's Faculty Hack meeting

MORE INFORMATION →
https://hackhpc.github.io/facultyhack-gateways23