Operating Systems Primer

Prerequisite for DS255

OS-Primer

- Operating Systems: Three easy pieces

  http://pages.cs.wisc.edu/~remzi/OSTEP/

Critical Facets to OS Functions

- Virtualization
- Concurrency
- Persistence

Virtualization Basics: Processor

- CPU/Processor as a resource and how it is used
  - Single program execution
  - Multi-programming and single program execution
  - Multi-tasking and multiple program execution
- Concept of single CPU and multiple processes
  - Abstraction of process and why it is necessary
  - How the process abstraction enables different use-cases of the CPU resource

Virtualization Basics: Memory

- Memory model to support program execution
  - Single program direct addressing
  - Multiple programs and relocatable addresses
  - Process abstraction and address space model
  - Memory segments and paging ideas
  - Support for paging constructs from hardware and OS

Virtualization Basics: I/O Subsystems

- Concept of synchronous and asynchronous communication
- I/O File abstraction and virtualization of I/O devices
- Support for access to I/O device
  - Hardware model for I/O devices
  - OS view of I/O devices
- Idea of buffering
  - Disk data
  - Network Data
  - Visualization Data

Concurrency Basics

- What does concurrency mean?
- Why do we need concurrency?
- How does the system deal with concurrency?
- Hardware constructs to support concurrency and how they are exposed and used by applications?
  - Processor: Does processor support concurrency?
  - Memory: How about memory?
  - I/O Devices: Are I/O Devices concurrent?
- OS constructs for concurrency

Hardware constructs to support concurrent processing

- How is hardware support built to enable such CPU usage?
- Contemporary processor architectures and concurrency capabilities.

OS features to use Processor concurrently

- Process Schedulers:
  - Single processor multiple processes
  - Multiple processors multiple processes
- Notion of time-sharing
  - Constructs required to support time-sharing
- Process Schedulers and concurrency

Hardware constructs to support concurrent memory access

- Hardware support for concurrent memory access
  - MMU
  - Processor-TLB

OS features to use memory concurrently

- Idea of process address spaces
- Protection and address spaces for process model.
- Support for isolation through segments and pages.
- Process page tables

Hardware features for concurrent I/O

- Traditional system design support for I/O
  - Single user, serial access
  - Privileged instruction access

OS features for concurrent I/O

- OS abstraction for I/O devices
  - Concept of file for device access
    - Serial and block access devices through files
    - Sockets
    - Pipes
    - Display devices
  - System call interface for I/O access
  - Asynchronous communication features

Persistence Basics

- What does persistence mean?
- Importance of persistence for system operations
- How is persistence requirement met in traditional systems?
- Visibility and accessibility of persistent data
- Varied devices with a reason for single view/API for access?

Hardware features for persistent data

- What are the ways in hardware to access persistent data devices?
- The design model and abstraction needed to use and access these devices.
- Concept of programmed I/O (PIO) instructions and memory mapped devices.
- Why do we need device drivers?
- How about concurrent device access and data channels?

OS view of Persistence

- Abstraction of devices into file and filesystem APIs
- Issues of consistency and integrity
- Synchronous/Asynchronous communication to devices
- Cost of abstraction for data access.
- System calls for device access

Word of caution!

- This is an advanced OS course so it is necessary that you should have taken an under-grad course on OS.
- This primer is for recap purposes and if you think these ideas are something you do not know then you should think of dropping this course!

Questions?

Thankyou!