Introduction to Operating Systems

References: OSTEP Ch-2

Syllabus

- Module A : Processes : Introduction, Abstraction, System Calls, Process Execution, Scheduling Policies, Inter-process Communication
- Module B: Memory: Introduction to Virtual Memory, Address Translation, Paging, Demand Paging, Memory Allocation and Free Space Algorithms
- Module C: Concurrency: Threads, Locks, Condition variables, Semaphores, bugs
- Module E: I/O and Filesystems: Communication with I/O, files and directories, files system implementation, hard disk internals
- Module F: Networking Subsytems: Network I/O using Sockets, Network I/O sub-system in Linux

Textbooks

- OSTEP - Operating Systems: Three Easy Pieces (OSTEP)- Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau
  - https://pages.cs.wisc.edu/~remzi/OSTEP/
- Xv6 – A simple, Unix-like teaching Operating System- xv6- Russ Cox, Frans Kaashoek, Robert Morris
  - https://pdos.csail.mit.edu/6.828/2018/xv6/book-rev11.pdf

References

- Operating System Concepts:8th Edition Wiley Student -Abraham Silberschatz, Peter Baer Galvin, Greg Gagne

Timing

- Theory - C1 - Classroom 5G2
  - Monday - 3:00 - 3:55 pm
  - Tuesday - 2:00 - 2:55 pm
  - Thursday - 5:00 - 5: 55pm (Reserved for Extra Classes and Quizzes)
  - Friday – 4:00 - 4:55 pm
- Lab - ML3
  - Wednesday - 9:00-11:55 am

Assessment

- Quiz-1 - Before Midsem
- Midsem- 18-09-2024
- Quiz-2 - Before End-Sem
- Endsem- 19-11-2024

Operating System

- Middleware
  - Sits between user program and Hardware
- Manages hardware:
  - CPU
  - Main memory
  - I/O Devices

Program Execution

- Compiler
  - Translated high level program (.c) into an executable (a.out)
  - The exe contains instructions that the CPU can understand, and data of the program (all numbered with addresses)
- Instructions run on CPU: hardware implements an instruction set architecture (ISA)
- CPU also consists of a few registers, e.g.,
  - Pointer to current instruction (program counter or PC or Instruction Pointer or IP)
  - Registers to hold operands of instructions and memory addresses

During Program Execution

- To run an exe, CPU
  - fetches instruction pointed at by PC from memory
  - loads data required by the instructions into registers
  - decodes and executes the instruction
  - stores results to memory
- Most recently used instructions and data are in CPU caches for faster access

What is the role of the OS?

- OS manages program memory
  - Loads program executable (code, data) from disk to memory
- OS manages CPU
  - Initializes program counter (PC) and other registers to begin execution
- OS manages external devices
  - Read/write files from disk

OS manages CPU

- OS provides the process abstraction
  - Process: A program in execution
  - OS creates and manages processes
- Each process perceives that it has exclusive access to the CPU; the OS virtualizes the CPU.
- Timeshares CPU among processes
- Enables coordination among processes

OS manages memory

- The operating system oversees the memory allocation for processes, including code, data, stack, heap, and other components.
- Each process operates under the assumption that it possesses an exclusive memory area, where code and data are assigned starting from address 0 (virtual addresses).
- The operating system shields the processes from the intricacies of physical memory management by handling the translation of virtual addresses to actual physical addresses.

OS manages devices

- OS has code to manage disk, and other I/O devices through device drivers
- Device drivers communicate directly with hardware devices using their specific protocols.
  - Issue commands to devices, such as fetching data from a file.
  - Handle interrupt signals from devices, such as when a user presses a key on the keyboard.
- Persistent data organized as a filesystem on disk

Goals to design an Operating System

- Conveniency- Provide abstraction of hardware resources for user programs

- Efficient
  - In terms of CPU utilization, memory, etc.

- Isolation
  - Among processes: One process does not overwrite other process

History of Operating Systems

- Initially incorporated as Libraries (set of functions)
  - Library - Provide common functionalities across programs
- Evolved as System Call
- What is a system call?
  - System call runs OS code, basically it means that CPU executes the code at a higher privilege level
- Earlier system executes a single program, whereas OS facilitates running multiple processes concurrently

Thank You