Operating Systems Notes

Sam Kirk

July 2024

Acronym Table

- POSIX: Portable Operating System Interface. a set of standard operating system interfaces based on the Unix operating system
- PID: Process Identifier.
- MMU: Memory Management Unit.
- PCB: Process Control Block. A structure that contains information about each process
- MLFQ: Multi Level Feedback Queue
- CFS: Completely Fair Scheduler
- PTE: Page Table Entry
- VPN: Virtual Page Number
- PFN: Page File Number
- DMI: Direct Media Interface
- PCIe: Peripheral Component Interconnect Express
- PIO: Programmed IO
- ISR: Interrupt Service Routine
- DMA: Direct Memory Access
- RAID: Redundant Array of Inexpensive Disks

Skipped Textbook Chapters

- 10
- half of 17
- Most of 23
- Chapter 27 is confusing

- Chapter 31 is confusing
- 33
- 36 is confusing
- Should probably read 37 again

Things to work on

- Address translation with paging (chapter 18)

Week 1 (Whole lecture made no sense)

What is an Operating System?

- A software that converts hardware into a usefull form for applications
- A resource allocator and control program making efficient use of hardware and managing the execution of user programs.

What does an OS provide?

- Abstraction - provides a standard library for resources
- (Resources are anything valuable, i.e. CPU, memory, disk)

Advantages of Abstraction

- Allows applications to reuse common facilites
- Make different devices look the same
- Provides higher-level or more useful functionality

What is a process?

- An execution stream in the context of a process state
- What is an execution stream?
- Stream of executing instructions
- Running piece of code
- Thread of control

Week 2

Status Bit

Determines if we are in user mode or kernel mode.

Scheduler

Determines the order that tasks are completed. Some common performance metrics for the scheduler are:

- Turnaround time: completion-time - arrival-time
- Response time: initial-schedule-time - arrival-time
- Waiting time: How long tasks spend in the ready queue
- Throughput: Jobs completed per unit of time
- Resource utilization: Keep expensive decives busy
- Overhead: The number of context switches
- Fairness: All jobs get the same amount of CPU over some time interval

Schedulers can be First Come First Served (FCFS), Shortest Job First (SJF), Shorted Time to First Completion (STCF), RR, Multi Level Feedback Queue (MLFQ), Completely Fair Scheduler (CFS).

The Dispatcher performs context-switches, i.e. switching from user mode to kernel mode.

In FCFS scheduling, convoys of small tasks tend to built up when a large one is running.

Preemptive Scheduling

Potentially schedule different jobs at any point by taking CPU away from running job. So we can run half a task and then if a shorter task comes along we can switch and run that task.

Prac Assignment 1

even.c

- fflush(stdout); flushes the output and prints what we have. Without this command it prints everything at the end of the loop.
- SIGINT is an interupt. Triggered by entering ctrl + c.
- SIGHUP is a signal hang-up. Need process PID number to execute. run ”kill -SIGHUP (PID number)” in another terminal to interupt.
- Get process PID using printf(”Process PID: % d”, getpid());

minishell.c

Included libraries and their purpose:

- ¡sys/types.h¿ and ¡sys/wait.h¿: For process control functions and types.
- ¡stdio.h¿: For input and output functions.

- ¡string.h¿: For string manipulation functions.
- ¡unistd.h¿: For POSIX operating system API functions.
- ¡stdlib.h¿: For general utilities, including memory allocation and process control.
- ¡signal.h¿: For signal handling.

Questions:

- What is a fork and how does it work?
- What is a perror
- What is the execvp function
- How many PIDs can run at once and are they related to the number of cores in the computer

Parent and Child Processes:
A parent process is a process that creates one or more child processes. The parent process typically performs operations that involve managing, monitoring, or coordinating with its child processes. It may create child processes to perform specific tasks concurrently or handle multiple tasks simultaneously.
A child process is a process that is created by a parent process. The child process executes instructions independently of the parent process. It inherits some properties from the parent process, such as open file descriptors, environment variables, and sometimes memory space. However, it has its own unique process ID (PID) and may execute different code or perform different tasks.
Possible Return Values from fork()

- If fork() returns a positive integer, this value is the process ID (PID) of the newly created child process
- If fork() returns 0, this indicates that the code is executing in the child process
- If fork() returns -1, it indicates that an error occurred while attempting to create a new process.

Week 3

Managing Processes with Base and Bounds

Give each process a base register for its memory use and give it a bound which it cannot exceed. Protection is required so a user cannot change the base and bound registers. Also a user cannot change to privileged mode. Some advantages of this Base and Bound resgisters are:

- Provides memory Protection
- Support dynamic relocation for process memories
- Simple and inexpensive (few registers, little MMU logic)

- Fast (add and compare occur in parallel)

Disadvantages are:

- Each process has a certian amount of space so this might go unused
- Also this memory is continuous (so its in one big block/list)
- Cannot share memory with other processes

Paging

Divide address spaces and physical memory into fixed-size pages.

Week 4

Week 5

Week 6