

CSE 134 - Embedded OS

Elijah Hantman

- What is operating system

Abstraction and resource manager. Provides clean beautiful APIs for ugly hardware.
Time sharing + space sharing, (CPU + memory)

- Computer Architecture

CPU, Memory, I/O.

I/O includes printers, screens, disk, any thing that isn't memory or CPU which the computer can interact with.

What is an interrupt? What is DMA?

Interrupts are hardware signals which force the CPU to execute specific instructions. They are used because sometimes we need to notify the CPU of an event.

Different modes. Polling (I/O request), Interrupts (Interrupt pins), DMA (direct memory access).

- DMA allows for no CPU usage, high throughput. Large data, ex: screens, GPUs, etc.
- Polling, requires no hardware support but requires busy-waiting, and is blocking.
- Interrupts do not require busy waiting but are low throughput, force a context switch, and requires specific hardware.
Software can also use interrupts, these are called system calls, and force the CPU to begin running kernel code in kernel mode.

CPU pipelining, improves throughput but not latency. Allows for temporal parallelism which allows heterogeneous tasks to occur simultaneously.

Use a buffer?! Idk man she doesn't explain very well.

Handling interrupts

Switches to kernel mode, uses a table to jump to interrupt handler. It then runs the interrupt code, which promises to save the CPU state (registers + memory). Then it runs in kernel mode until eventually it switches back into the process and restores the CPU state. This is also when the CPU scheduler will run for timesharing. I/O syscalls will cause the thread to be blocked and it will wait in a queue until it is ready to run again.

- Processes

- Process is a program in execution

Data, memory, file descriptors.

Binary program, instruction stream. Status (Running, Ready, Blocked), stack, which is part of the memory of the program.

It also has other resources which in Linux are usually represented by file descriptors.

Memory means an address space, which is a mapping from the addresses emitted by the running process to physical addresses which can be interpreted by the physical memory.

Address space = location in memory.

- What is a thread?

Threads are CPU execution states. Threads have their own stacks but share an address space with their parent process.

Thread context switches are cheaper since it doesn't have to switch the page table.

Running to blocked when I/O (polling or interrupt)

Blocked to Ready when I/O finishes (polling or interrupt)

Ready to Running when Scheduled

Running to Ready when non blocking request?
Running to Ready when preemption (round robin)

– What is a context switch

When the CPU changes from a running process to another process. It also includes switches from User space to Kernel mode.

– Scheduler Metrics

Utilization

$$U = 1 - (\prod u_i)$$

Waiting Time: Queue waiting time

This is the amount of time spent waiting in the ready queue.

Turnaround time: the time from the start of a process to its termination.