The fundamental role in process management is keeping track of the processes, their states

Process Table

PCB0
PCB1
PCB2
PCB3
PCB4

Each process control block (PCB) contains several bytes of information, as discussed in yesterday's notes.

Why store accounting info?

OS Scheduling algorithm can determine whether processes are given enough / too much time.

OS11 PCB

SP_{High}
SP_{Low}

OS11 Process Table

$SP0_{Hi}$
$SP0_{Lo}$
$SP1_{Hi}$
$SP1_{Lo}$
$SP2_{Hi}$
$SP2_{Lo}$
$SP3_{Hi}$
$SP3_{Lo}$

8 bytes

MiniIDE (Motorola Basic Assembly)

```
DATA            EQU             0x8600

                ORG             DATA

PROCTBL:        DS.B            8

                ORG             CODE

                LDX             #PROCTBL        ; Load X with the number PROCTABLE
```

Immediate Addressing Mode

Load the immediate number
Since PROCTBL is a label, its number is its address

OS Scheduler

Chooses processes to move into the CPU.

An OS can be designed so that processes can be queued for loading.

Three components:

1. Long-term scheduler
    a. Choose load → ready behavior
    b. In a limited memory system, we can't have all of the processes that want to run currently in memory at once. Instead, those requesting load.
    c. The long-term scheduler determines which can move from the load queue to the ready queue. The performance of the long-term scheduler is no nearly as critical as the performance of the other two because its task is less periodic.
    d. In Windows, Unix, MacOS, etc., probably don't have long-term schedulers in their software code. However, this is a big concept on batch OSes, in which hundreds of jobs are requested and an immediate response is not necessary. We don't have batch OSes, but user level OSes in which we want immediate responses.
    e. **Balances between I/O bound and CPU bound processes**
2. Medium-term scheduler
    a. Choose processes to swap **in and out** of disk file
    b. Memory is a limited resource on our computer system. The main memory system is fixed. The medium-term scheduler stores a process's active state and stores it to a swap file on the disk when the system resources demand it. The process is then suspended on the disk.
3. Short-term scheduler
    a. The CPU scheduler (context switcher). Moves processes from ready to running.
    b. Most frequently run scheduling algorithm, therefore is the **most critical performance.** Time to do the context switch (**dispatch latency**) should take no longer than 5% of the time slice.

Time slice = dispatch latency + process execution time

The medium and short term schedulers tend to be combined together in the kernel behavior.

Process Characteristics

1. CPU-bound
    a. Spends the majority of its time on the CPU and uses I/O infrequently.
2. I/O-bound
    a. Spends the majority of its time waiting for I/O and uses the CPU infrequently.

Processes do: CPU-burst → I/O burst → repeat

Statistically, most processes have a 2-3 ms burst time before they wait for something. This understanding helps us to set the appropriate time slice value