COA

Chapter 08: | OS Support – OS Types

Prof. Dr. Mostafa Elhosseini

https://youtube.com/drmelhosseini

AGENDA

- Types of OS

Types of Operating Systems

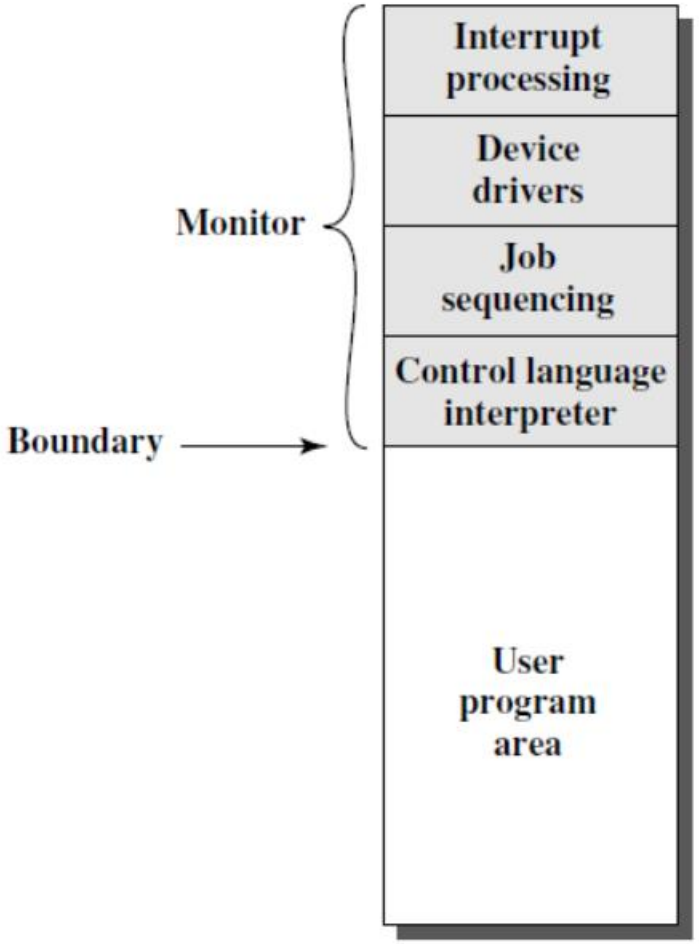
- Interactive
  - User/programmer **interacts directly with the computer**
- Batch
  - The user's program is **batched together** with programs from other users and submitted by a computer operator
  - Pure batch systems are **rare today**
- Single program (Uni-programming)
  - only one program at a time
- Multi-programming (Multi-tasking)
  - keep the processor **as busy as possible**, by having it work on **more than one program** at a time. Several programs are **loaded** into memory, and the processor **switches rapidly among them**

Early Systems

- Late 1940s to mid 1950s
- No Operating System
- Programs interact directly with hardware
  - The programmer could proceed to examine registers and main memory to determine the cause of the error
- Two main problems:
  - Scheduling
  - Setup time

Simple Batch Systems - Monitor

- Early Systems has **wasted time due to scheduling** and **setup time** was **unacceptable**
  - Maximize processor utilization
- **Resident** Monitor program
  - Portion of monitor resident in memory
- Users submit jobs to operator
  - the user no longer has direct access to the processor
- Operator batches jobs
- Monitor controls sequence of events to process batch
- **When one job is finished, control returns to Monitor which reads next job**
- Monitor handles scheduling



Job Control Language

- Instructions to Monitor
- Usually denoted by $
- e.g.

```
$JOB
$FTN
... Some Fortran instructions
$LOAD
$RUN
... Some data
$END
```

Desirable Hardware Features

- Memory protection
  - To protect the Monitor
- Timer
  - To prevent a job monopolizing the system
- Privileged instructions
  - Only executed by Monitor
  - e.g. I/O
- Interrupts
  - Allows for relinquishing and regaining control

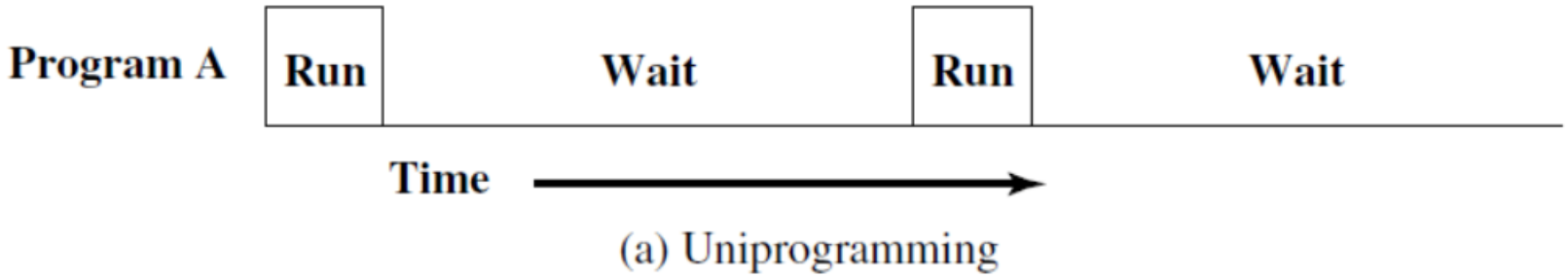
MULTIPROGRAMMED BATCH SYSTEMS

- **Even** with the automatic job sequencing provided by a simple batch OS, **the processor is often idle**. The problem is that **I/O devices are slow** compared to the processor
- The processor spends a certain amount of time executing, until it **reaches an I/O instruction**. It must then **wait** until that I/O instruction concludes before proceeding
  - When one program is waiting for I/O, another can use the CPU

Read one record from file	15 μs
Execute 100 instructions	1 μs
Write one record to file	15 μs
TOTAL	31 μs

$$\text{Percent CPU utilization} = \frac{1}{31} = 0.032 = 3.2\%$$

MULTIPROGRAMMED BATCH SYSTEMS

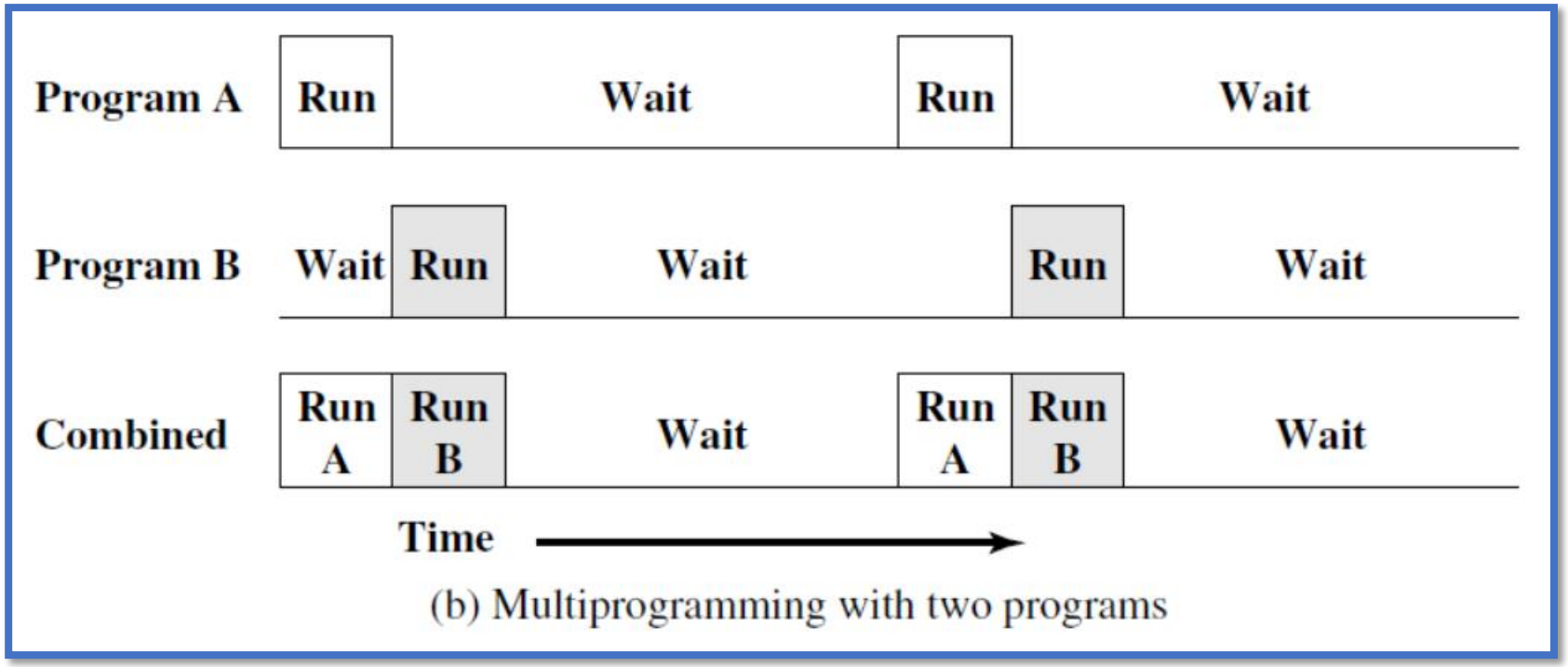


(a) Uniprogramming

- the processor can **issue an I/O command** for one job and **proceed with** the execution of **another** job while the I/O is carried out by the device controller. When the **I/O operation** is **complete**, the processor is **interrupted,** and control is passed to an **interrupt-handling program** in the OS

MULTIPROGRAMMED BATCH SYSTEMS

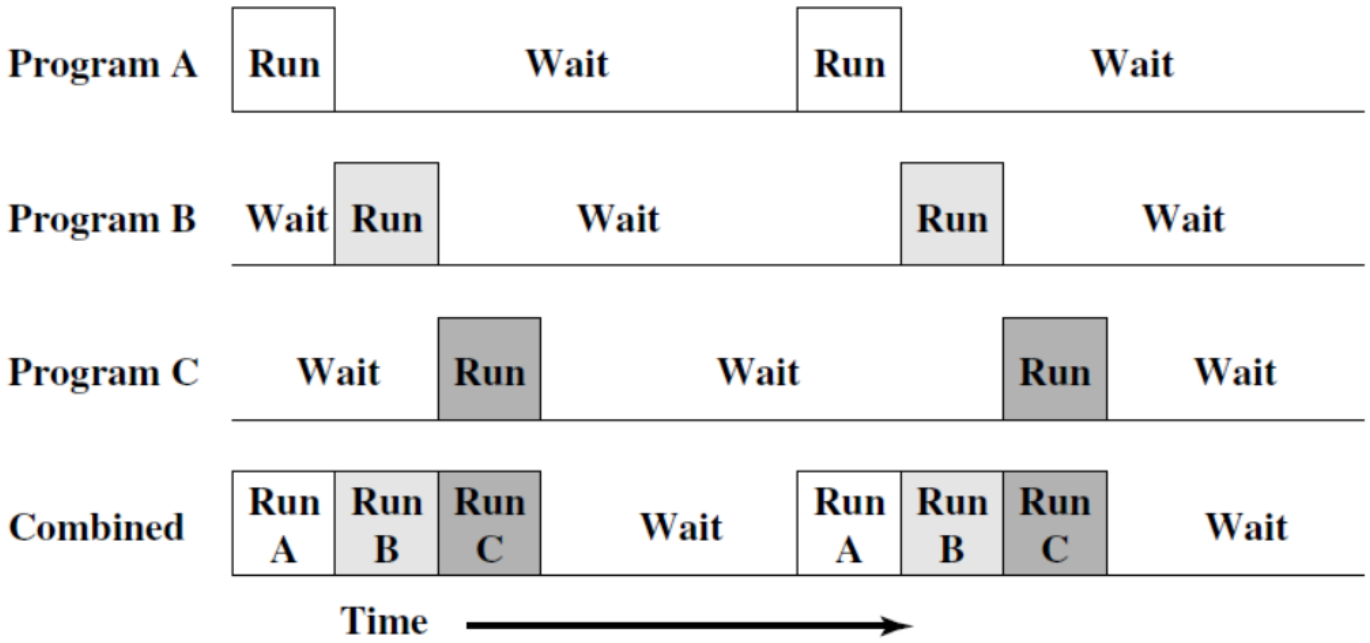
- When one job needs to wait for I/O, the processor can switch to the other job, which likely is not waiting for I/O



(b) Multiprogramming with two programs

MULTIPROGRAMMED BATCH SYSTEMS

- This technique is known as **multiprogramming**, or **multitasking**. It is the central theme of modern operating systems.



(c) Multiprogramming with three programs

Multiprogramming

- **Multiprogramming** operating systems are fairly sophisticated compared to single program, or uni-programming, systems.
- To have several jobs ready to run, the **jobs must be kept in main memory**, requiring some form of **memory management**. In addition, if several jobs are ready to run, the processor must **decide which one to run**, which requires some algorithm for **scheduling**

Time-sharing Systems

- Allow users to interact directly with the computer
- i.e. Interactive
- handle multiple interactive jobs
- processor’s time is shared among multiple users.
- In a time-sharing system, multiple users simultaneously access the system through terminals