Computer Science CSCI 251

Systems and Networks
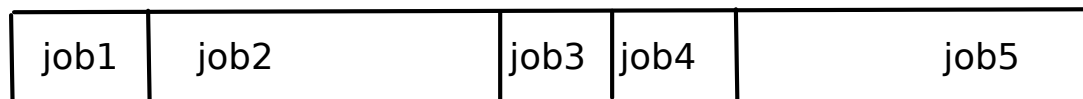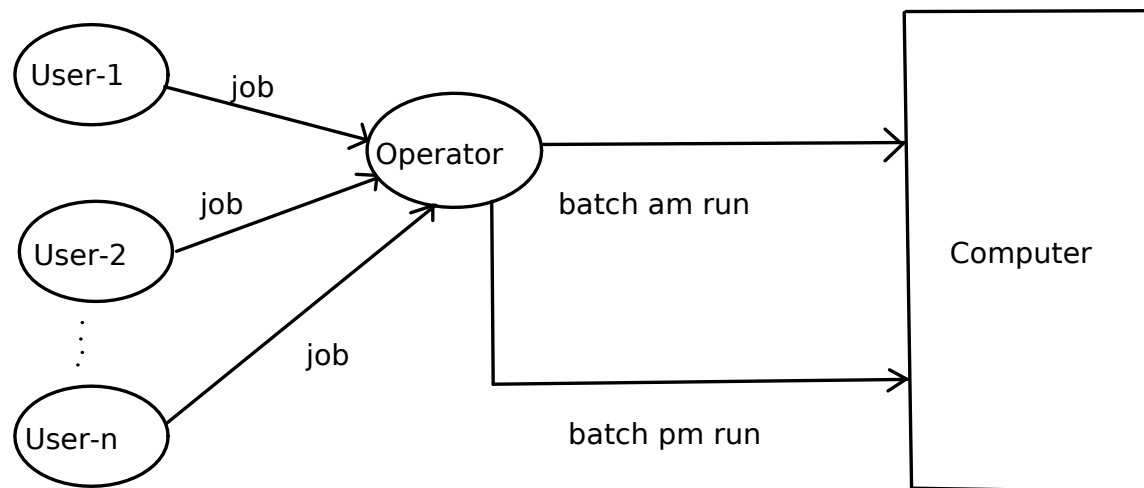
Dr. Peter Walsh

Department of Computer Science

Vancouver Island University

peter.walsh@viu.ca

Batch Processing (1950's)

Batch Processing (1950's) cont.

○ Behaviour

- once a job is running, it has total control
  of the computer
- jobs run to completion or terminate abnormally

○ Advantages

- increased performance since operator intervention
  is not required between jobs in the batch

○ Disadvantages

- no interaction between the user and his/her job
- what if a job had an infinite loop?

Batch Jobs With Input/Output (I/O)

Jobn

execute	wait for I/O	execute	wait for I/O	execute
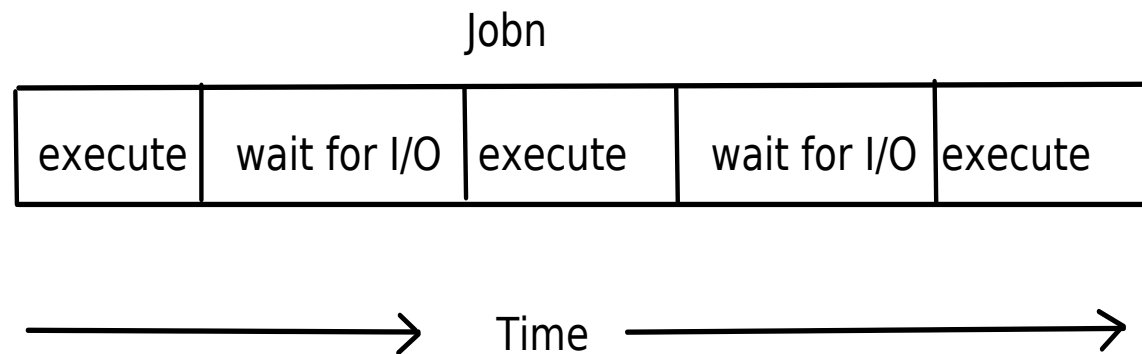
⟶ Time ⟶

○ Problem (I/O delays)
- a lot of time is wasted waiting for I/O

○ Solution (Multiprogramming)
- execution of many jobs concurrently
- switch the CPU from one job to another when the first becomes blocked while waiting for I/O

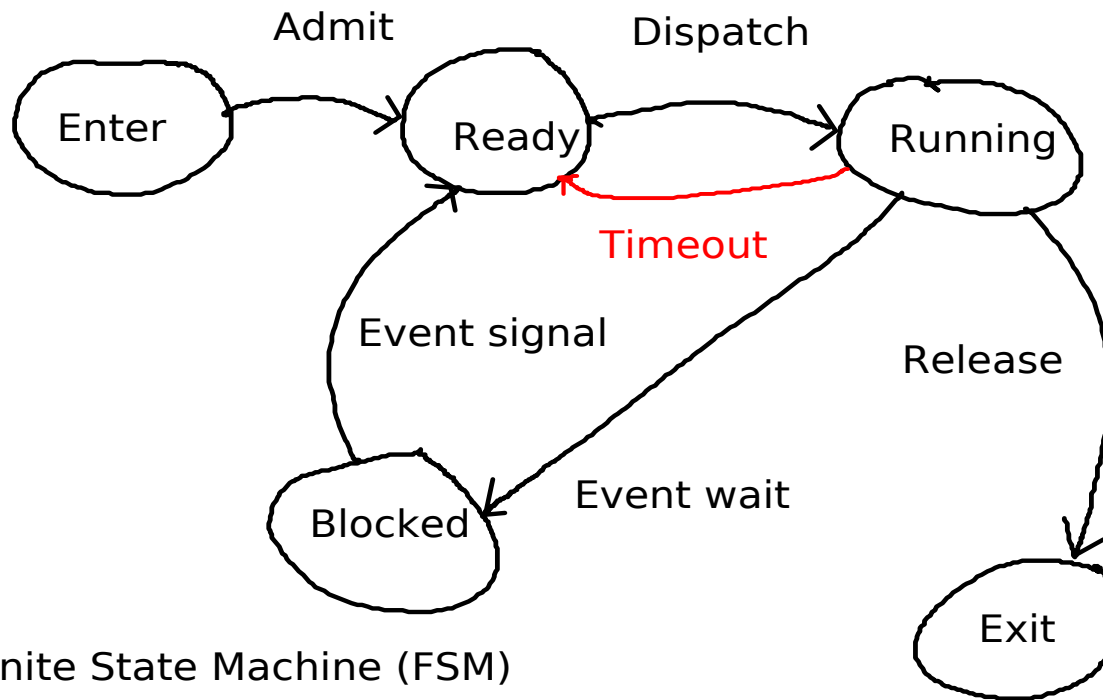# Multiprogramming Model

- a process is a program in execution
- the term job is replaced by process
- still no user interaction with processes

Finite State Machine (FSM)
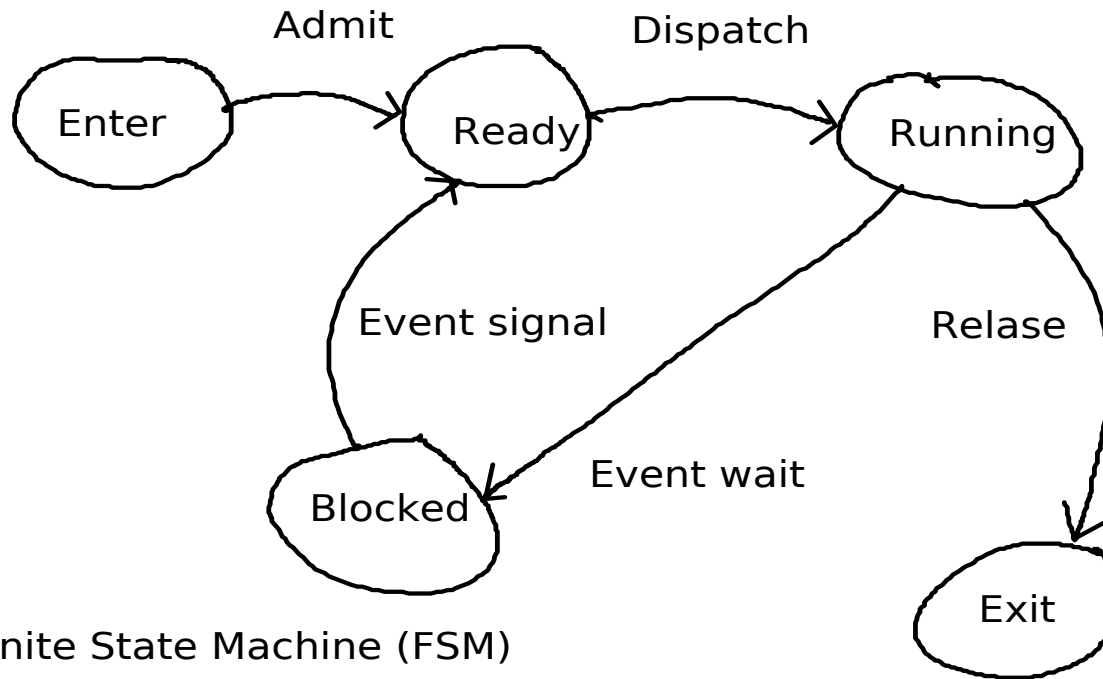
Timesharing Model

- also known as multitasking
- increases responsiveness

Finite State Machine (FSM)

Process Execution

○ Dispatcher
- responsible for moving a process from
  the **Ready** state to the **Running** state
- is concerned with the low-level mechanisms
  of running a process (context switch)

○ Scheduler
- responsible for selecting the next process
  to execute from processes in the **Ready** state
- is concerned with scheduling policies

Scheduling Policy Metrics

○ Processor Utilization

  • percentage of time a processor spends

    executing processes

○ Response Time

  • time from process release to the beginning

    of a response

○ Throughput

  • number of completed processes per unit time

○ Waiting Time

  • time a process waits in the **Ready** state

○ Turnaround Time

  • time from process release to completion

Unix History

○ https://en.wikipedia.org/wiki/History_of_Unix

○ Multics (Mid 1960's)
  - MIT, Bell Labs and General Electric
  - GE-645 mainframe computer

○ Unix (1973)
  - Bell Labs
  - DEC PDP mini computers
  - licensed it to educational institutions

○ Linux (1991)
  - rewrite Unix kernel for the x86 architecture
  - IBM PC, Home Computer

Linux

○ Linux Torvalds
- initial release 1991

Distribution

Userland	GNU Packages
	BuisyBox
X86 Kernel	
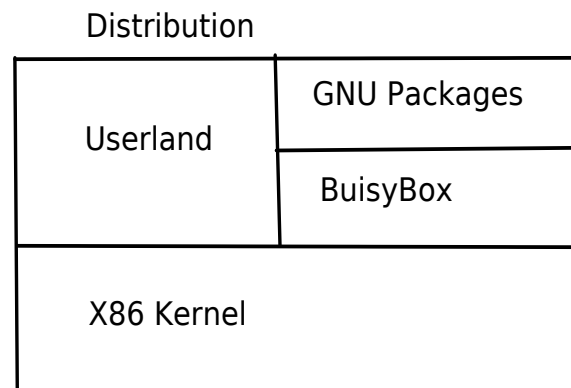
○ Distributions
- TinyCore
- Debian
- Ubuntu

Distribution Versions

○ TinyCore
- version 6.4 (buisybox), kernel 3.6 (2015)
- version 11.1 (buisybox), kernel 5.43 (2020)
- rolling release

○ Debian
- version 10 Buster (gnu), kernel 4.19 (2019)
- version 11 Bullseye (gnu) (TBD)
- scheduled release

○ Ubuntu
- version 20.04 LTS Focal Fossa (gnu),
  kernel 5.4 (April 2020)
- version 20.10 Groovy Gorilla (gnu), (October 2020)
- LTE (Long Term Support)

Bootable Linux Installation

○ Full Install
- • traditional installation where code and data
    are stored on disk(s) (secondary storage unit(s))
- • sometimes referred to as a scatter install

○ Live CD
- • OS boots and runs directly from CD
- • typically, there is no code nor data
    persistence and the system returns to
    its pre-boot state on shutdown
- • uses include full-install system rescue,
    repair and installation

TinyCore

○ Overview

- loads itself into RAM from storage
- small foot print (8MB - 64MB RAM)

○ Installation Options (CSCI 251 Labs)

- Live CD: boot from CD (iso)

  uses ISOLINUX as the boot-loader

- Frugal: system in two files

  uses GRUB as the boot-loader

- Hard Drive: scatter mode

  uses GRUB as the boot-loader

TinyCore Materials

○ Live CD

 • /srv/export/TinyCore-current.iso
   (file on otter and the cubs)

 • http://csci.viu.ca/export/TinyCore-current.iso
   (available through the web)

○ Virtual Hard Disk

 • download from the course web page

○ On VIU Equipment

 • each student shares one copy of the Live CD iso

 • each student had their own copy of the
   virtual hard drive stored in ∼/VirtualBox VMs