CSGE602055 Operating Systems
CSF2600505 Sistem Operasi
Week 08: Scheduling

Rahmat M. Samik-Ibrahim (ed.)

University of Indonesia

https://os.vlsm.org/
Always check for the latest revision!

REV235 13-Sep-2020

Operating Systems 2020-2 – (A, I, M) **from HOME** A/M [Tu 10-12, ZOOM] — I [Tu 08-10, ZOOM]

Week	**Schedule**	**Topic**	**OSC10**
Week 00	15 Sep - 21 Sep 2020	Overview 1, Virtualization & Scripting	Ch. 1, 2, 18.
Week 01	22 Sep - 28 Sep 2020	Overview 2, Virtualization & Scripting	Ch. 1, 2, 18.
Week 02	29 Sep - 05 Oct 2020	Security, Protection, Privacy, & C-language	Ch. 16, 17
Week 03	06 Oct - 12 Oct 2020	File System & FUSE	Ch. 13, 14, 15
Week 04	13 Oct - 19 Oct 2020	Addressing, Shared Lib, & Pointer	Ch. 9
Week 05	20 Oct - 26 Oct 2020	Virtual Memory	Ch. 10
Week 06	27 Oct - 16 Nov 2020	Concurrency: Processes & Threads	Ch. 3, 4
	29 Oct 2020	Maulid Nabi	
Week 07	17 Nov - 23 Nov 2020	Synchronization & Deadlock	Ch. 6, 7, 8
Week 08	24 Nov - 30 Nov 2020	Scheduling + W06/W07	Ch. 5
Week 09	01 Dec - 07 Dec 2020	Storage, Firmware, Bootldr, & Systemd	Ch. 11
Week 10	08 Dec - 16 Dec 2020	I/O & Programming	Ch. 12
	09 Dec 2020	Pil Kada	

STARTING POINT — https://os.vlsm.org/

- **Text Book** — Any recent/decent OS book. Eg. (**OSC10**) Silberschatz et. al.: **Operating System Concepts**, 10^{th} Edition, 2018. See also http://codex.cs.yale.edu/avi/os-book/OS10/.
- **Resources**
  - **Extra Scele from Home** — https://scele.cs.ui.ac.id/course/view.php?id=3020.
  - **Download Slides and Demos from GitHub.com** https://github.com/UI-FASILKOM-OS/SistemOperasi/
  - **Problems** — https://rms46.vlsm.org/2/: 195.pdf (W00), 196.pdf (W01), 197.pdf (W02), 198.pdf (W03), 199.pdf (W04), 200.pdf (W05), 201.pdf (W06), 202.pdf (W07), 203.pdf (W08), 204.pdf (W09), 205.pdf (W10).
- **Build your own Virtual Guest** https://osp4diss.vlsm.org/

Agenda

Week 08 Scheduling: Topics[1]

- Preemptive and non-preemptive scheduling
- Schedulers and policies
- Processes and threads
- Deadlines and real-time issues

[1] Source: ACM IEEE CS Curricula 2013

Week 08 Scheduling: Learning Outcomes[1]

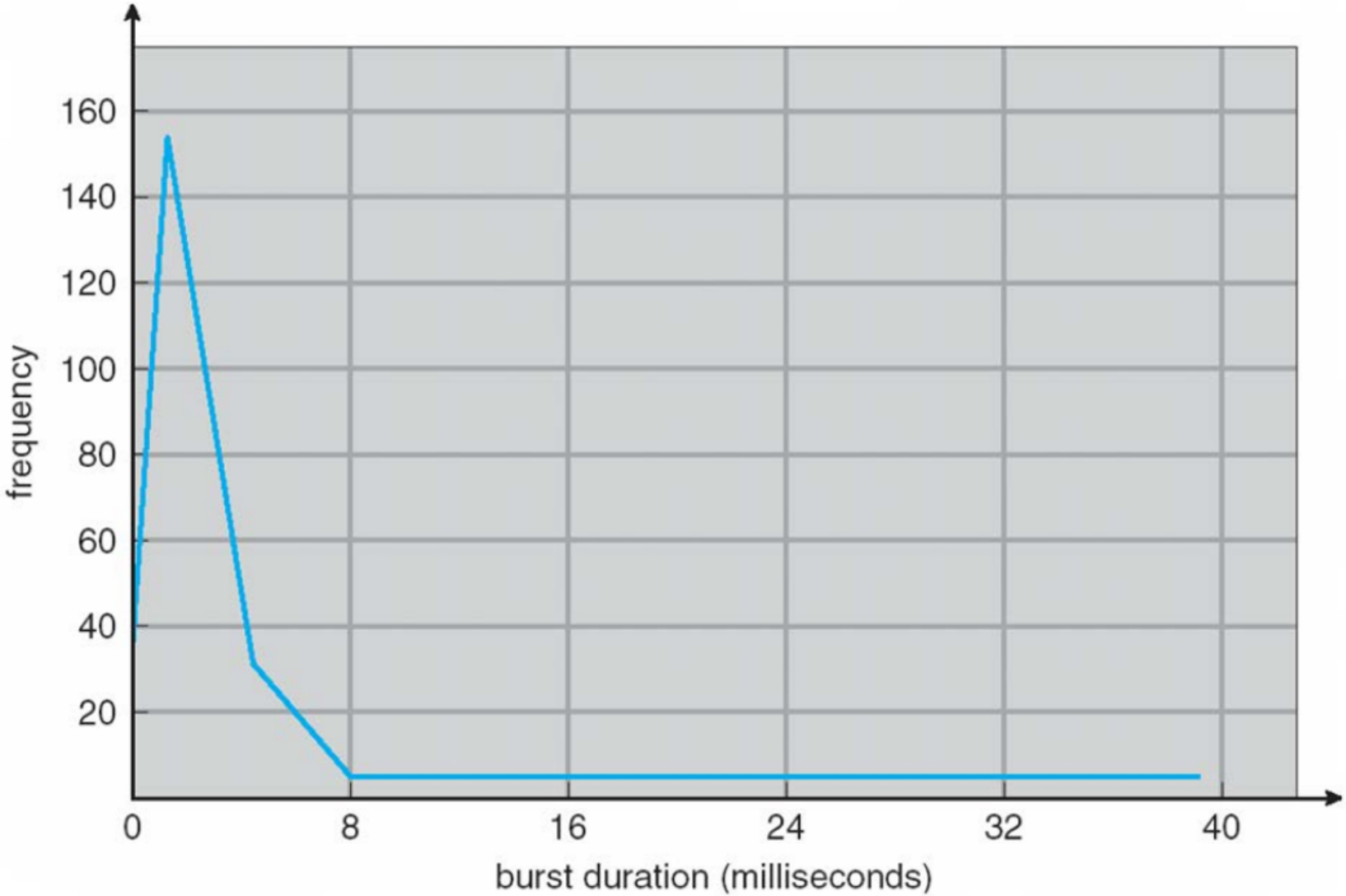
- Compare and contrast the common algorithms used for both preemptive and non-preemptive scheduling of tasks in operating systems, such as priority, performance comparison, and fair-share schemes. [Usage]
- Describe relationships between scheduling algorithms and application domains. [Familiarity]
- Discuss the types of processor scheduling such as short-term, medium-term, long-term, and I/O. [Familiarity]
- Describe the difference between processes and threads. [Usage]
- Compare and contrast static and dynamic approaches to real-time scheduling. [Usage]
- Discuss the need for preemption and deadline scheduling. [Familiarity]
- Identify ways that the logic embodied in scheduling algorithms are applicable to other domains, such as disk I/O, network scheduling, project scheduling, and problems beyond computing. [Usage]

[1] Source: ACM IEEE CS Curricula 2013

Week 08: Scheduling

- Reference: (OSC10-ch05 demo-w08)
- Scheduling
  - Basic Concepts
    - **WARNING:** It's just a BURST
    - IO Burst
    - CPU Burst
    - CPU Burst vs. Freq (See next slide)
  - Criteria: Utilization, throughput, {turnaround, waiting, response} time.
  - (Burst) Algorithm
    - FCFS, SJF, RR, Priority, Multilevel Queue.
  - Preemptive / Non-preemptive (Cooperative) Scheduling
  - I/O Bound / CPU Bound Processes
- Thread Scheduling
  - User-level $\rightarrow$ Process-Contention Scope (PCS): many to many/one.
  - Kernel-level $\rightarrow$ System-Contention Scope (SCS): one to one.
- Standard Linux Scheduling
  - Completely Fair Scheduler (CFS).
  - Real Time Scheduling.

CPU Burst: How Long (When)?



MultiProcessor Schedulling

- Asymmetric Multiprocessing vs. Symmetric Multiprocessing (SMP).
- Processor Affinity: soft vs. hard.
- NUMA: Non-Uniform Memory Access.
- Load Balancing
- Multicore Processors
- Real Time Schedulling: Soft vs. Hard.
- Big O Notation
  - O(1)
  - O(log N)
  - O(N)

The Two State Model

- CPU State – I/O State – CPU State – ...
  - n: processes in memory.
  - p: I/O time fraction.
  - p^n: probability n processes waiting for I/O.
  - $1 - p^n$: CPU utilization of n processes.
  - $\left[\frac{(1-p^n)}{n}\right]$: CPU utilization of ONE processes.
- Example: $p = 60\% \Rightarrow$ **CPU Utilization Per Process**: $\left[\frac{1-(60\%)^n}{n}\right]$

CPU Utilization	**Multiprogramming (%)**				
N	**1**	**2**	**3**	**4**	**5**
Per Process	40	32	26	21	18

  - For 5 concurrent processes:
    If total time is 100 seconds; for each processs, the CPU time will be 18 seconds.

The End

- ☐ This is the end of the presentation.
- ☒ This is the end of the presentation.
- This is the end of the presentation.