CSGE602055 Operating Systems
CSF2600505 Sistem Operasi
Week 08: Scheduling

Rahmat M. Samik-Ibrahim

University of Indonesia

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

REV158 29-Aug-2018

Operating Systems 2018-2 (Room 3114)
R/M (Tu/Th 13-15) | I (Tu/Th 15-17) | E (Th 19-22)

Week	Schedule	Topic	OSC10
Week 00	04 Sep - 12 Sep 2018	Overview 1	Ch. 1, 18
Week 01	13 Sep - 19 Sep 2018	Overview 2 & Scripting	Ch. 1, 2
Week 02	20 Sep - 26 Sep 2018	Security, Protection, Privacy, & C-language	Ch. 16, 17
Week 03	27 Sep - 03 Oct 2018	File System & FUSE	Ch. 13, 14, 15
Week 04	04 Oct - 10 Oct 2018	Addressing, Shared Lib, & Pointer	Ch. 9
Week 05	11 Oct - 17 Oct 2018	Virtual Memory	Ch. 10
Reserved	18 Oct - 19 Oct 2018		
Mid-Term	20 Oct - 27 Oct 2018	MidTerm (UTS): TBA	
Week 06	30 Oct - 05 Nov 2018	Concurrency: Processes & Threads	Ch. 3, 4
Week 07	06 Nov - 12 Nov 2018	Synchronization & Deadlock	Ch. 6, 7, 8
Week 08	13 Nov - 21 Nov 2018	Scheduling	Ch. 5
Week 09	22 Nov - 28 Nov 2018	Disks, BIOS, Loader, & Systemd	Ch. 11
Week 10	29 Nov - 05 Dec 2018	I/O & Programming	Ch. 12
Reserved	06 Dec - 14 Dec 2018		
Final	15 Dec - 22 Dec 2018	Final (UAS): TBA	This schedule is subject to change.
Extra	12 Jan 2019	Extra assignment	

The Weekly Check List

- ☐ **Resources**: `https://os.vlsm.org/`
    - ☐ **(THIS) Slides** — `https://github.com/UI-FASILKOM-OS/ SistemOperasi/tree/master/pdf/`
    - ☐ **Demos** — `https://github.com/UI-FASILKOM-OS/ SistemOperasi/tree/master/demos/`
    - ☐ **Extra** — `BADAK.cs.ui.ac.id:///extra/`
    - ☐ **Problems** — `rms46.vlsm.org/2/195.pdf`, `196.pdf`, ..., `205.pdf`
- ☐ **Text Book**: any recent/decent OS book. Eg. (**OSC10**) Silberschatz et. al.: **Operating System Concepts**, 10th Edition, 2018.
- ☐ Encode your **QRC** with image size of approximately 250x250 pixels:
  "OS182 CLASS ID SSO-ACCOUNT Your-Full-Name"
  Special for **Week 00**, mail your **embedded** QRC to:
  `operatingsystems@vlsm.org`
  With Subject: `OS182 CLASS ID SSO-ACCOUNT Your-Full-Name`
- ☐ Write your Memo (with QRC) **every week**.
- ☐ Login to `badak.cs.ui.ac.id` via `kawung.cs.ui.ac.id` for at least **10 minutes** every week. Copy the weekly demo files to your own home directory.
  Eg. (Week00): `cp -r /extra/Week00/W00-demos/ W00-demos/`

Agenda

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

---

[1]Source: ACM IEEE CS Curricula 2013

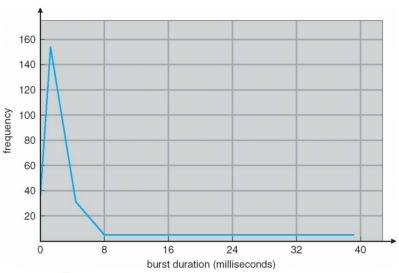# Week 08 Scheduling: Learning Outcomes[1]

- 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 → Process-Contention Scope (PCS): many to many/one.
    - Kernel-level → System-Contention Scope (SCS): one to one.
- Standard Linux Scheduling
    - Completely Fair Scheduler (CFS).
    - Real Time Scheduling.

CPU Burst: How Long (When)?

©2013 Silberschatz, Galvin and Gagne   Operating System Concepts – 9th Edition

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.