CSGE602055 Operating Systems
CSF2600505 Sistem Operasi
Week 08: Scheduling

Rahmat M. Samik-Ibrahim (ed.)

University of Indonesia

REV321 26-Jul-2021

Week	Schedule & Deadline[1])	Topic	OSC10[2])
Week 00	XX Xxx - XX Xxx 2021	Overview 1, Virtualization & Scripting	Ch. 1, 2, 18.
Week 01	XX Xxx - XX Xxx 2021	Overview 2, Virtualization & Scripting	Ch. 1, 2, 18.
Week 02	XX Xxx - XX Xxx 2021	Security, Protection, Privacy, & C-language.	Ch. 16, 17.
Week 03	XX Xxx - XX Xxx 2021	File System & FUSE	Ch. 13, 14, 15.
Week 04	XX Xxx - XX Xxx 2021	Addressing, Shared Lib, & Pointer	Ch. 9.
Week 05	XX Xxx - XX Xxx 2021	Virtual Memory	Ch. 10.
Week 06	XX Xxx - XX Xxx 2021	Concurrency: Processes & Threads	Ch. 3, 4.
Week 07	XX Xxx - XX Xxx 2021	Synchronization & Deadlock	Ch. 6, 7, 8.
Week 08	XX Xxx - XX Xxx 2021	Scheduling + W06/W07	Ch. 5.
Week 09	XX Xxx - XX Xxx 2021	Storage, Firmware, Bootloader, & Systemd	Ch. 11.
Week 10	XX Xxx - XX Xxx 2021	I/O & Programming	Ch. 12.

---

[1]) The **DEADLINE** of Week 00 is XX Xxx 2021, whereas the **DEADLINE** of Week 01 is XX Xxx 2021, and so on...

[2]) Silberschatz et. al.: **Operating System Concepts**, 10th Edition, 2018.

[3]) This information will be on **EVERY** page two (2) of this course material.

- ☐ **Text Book** — Any recent/decent OS book. Eg. (**OSC10**) Silberschatz et. al.: **Operating System Concepts**, 10th Edition, 2018. See also https://www.os-book.com/OS10/.
- ☐ **Resources**
    - ☐ **SCELE OS212** — https://scele.cs.ui.ac.id/course/view.php?id=XXXX. The enrollment key is **XXX**.
    - ☐ **Download Slides and Demos from GitHub.com** https://github.com/UI-FASILKOM-OS/SistemOperasi/: os00.pdf (W00), os01.pdf (W01), os02.pdf (W02), os03.pdf (W03), os04.pdf (W04), os05.pdf (W05), os06.pdf (W06), os07.pdf (W07), os08.pdf (W08), os09.pdf (W09), os10.pdf (W10).
    - ☐ **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).
    - ☐ **LFS** — http://www.linuxfromscratch.org/lfs/view/stable/
    - ☐ **OSP4DISS** — https://osp4diss.vlsm.org/
    - ☐ **DOIT** — https://doit.vlsm.org/001.html

Agenda

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

[1]Source: ACM IEEE CS Curricula 2013

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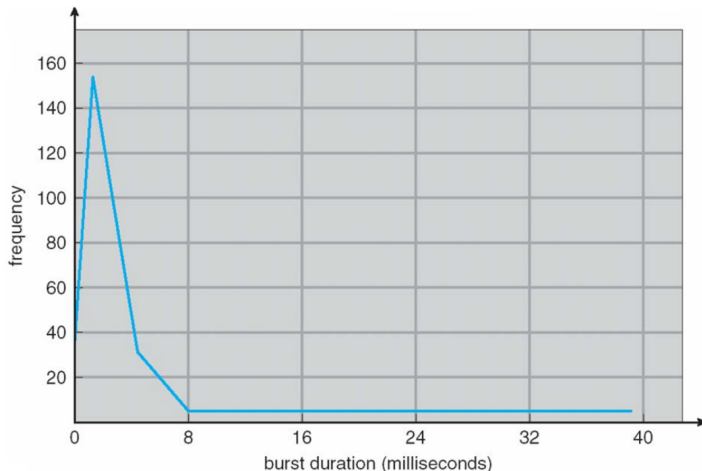
©2013 Silberschatz, Galvin and Gagne Operating System Concepts – 9th Edition

Figure: Burst: Duration vs Frequency

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.

- ☐ Week 08 Token: AkunGitHubW08 Eg. **cbkadalW08**
- ☐ This page is https://os.vlsm.org/Slides/check08.pdf.
- ☐ More details: https://osp4diss.vlsm.org/W08.html.
- ☐ Assignment Check List:
  1. Read: (OSC10 chapter 05)
  2. Visit https://os.vlsm.org/GitHubPages/. Review **Last Week TOP 10 List** and pick at least 3 out of your 10 next neighbors. See https://cbkadal.github.io/os212/TXT/myrank.txt. item Create your **TOP 10 List** of Week 08. **Do not use lecture material. Please be more creative!** (E.g. https://cbkadal.github.io/os212/W08/).
  3. Week 08 - 10 will be about building "Linux From Scratch (LFS)"
  4. There will be no download. The result of each week should be placed into a "W08/" folder and tarballed as "myW08.tar.bz2.asc"
  5. Update your log (e.g. https://cbkadal.github.io/os212/TXT/mylog.txt).
  6. Make SHA256SUM and sign it (detached, armor) as SHA256SUM.asc.

The End

☐ This is the end of the presentation.

☒ This is the end of the presentation.

- This is the end of the presentation.