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!

REV271 28-Feb-2021

Operating Systems 211[3]) — **PJJ from HOME** ZOOM: A [Mon (or Wed) 10:00] — B [Mon (or Wed) 15:40] — C [Tue (or Thu) 08:00]

Week	**Schedule & Deadline**[1])	**Topic**	**OSC10**[2])
Week 00	22 Feb - 28 Feb 2021	Overview 1, Virtualization & Scripting	Ch. 1, 2, 18.
Week 01	01 Mar - 07 Mar 2021	Overview 2, Virtualization & Scripting	Ch. 1, 2, 18.
Week 02	08 Mar - 16 Mar 2021	Security, Protection, Privacy, & C-language.	Ch. 16, 17.
Week 03	17 Mar - 23 Mar 2021	File System & FUSE	Ch. 13, 14, 15.
Week 04	24 Mar - 30 Mar 2021	Addressing, Shared Lib, & Pointer	Ch. 9.
Week 05	31 Mar - 06 Apr 2021	Virtual Memory	Ch. 10.
Week 06	07 Apr - 25 Apr 2021	Concurrency: Processes & Threads	Ch. 3, 4.
Week 07	26 Apr - 02 May 2021	Synchronization & Deadlock	Ch. 6, 7, 8.
Week 08	03 May - 09 May 2021	Scheduling + W06/W07	Ch. 5.
Week 09	17 May - 23 May 2021	Storage, Firmware, Bootloader, & Systemd	Ch. 11.
Week 10	24 May - 06 Jun 2021	I/O & Programming	Ch. 12.

[1]) The **DEADLINE** of Week 00 is 28 Feb 2021, whereas the **DEADLINE** of Week 01 is 07 Mar 2021, and so on...

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

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

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**
  - **SCELE OS211** — https://scele.cs.ui.ac.id/course/view.php?id=3134. 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

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)?

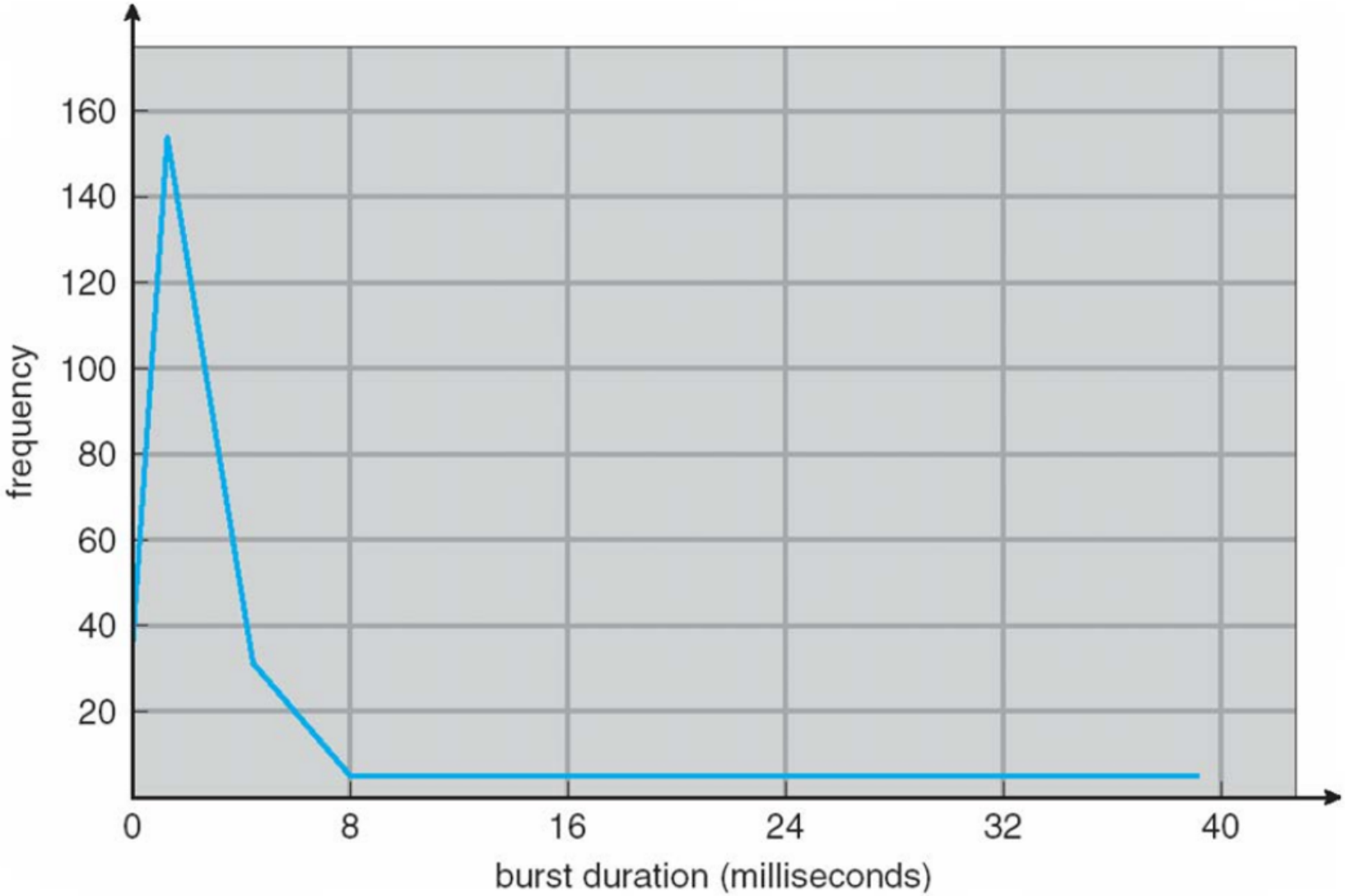

Figure: Burst Duration vs Frequency

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.

Linux From Scratch

- Week 08 (part 01).
  - Create a blank OVA file for a Debian Guest (uninstalled):
    - File/signature name: "LFSA.ova"/"LFSA.ova.asc" (armor, detach).
    - RAM: 1GB or better.
    - SATA Disk#1 (16 GB): Formated for root "/".
    - SATA Disk#2 (32 GB): Formated for LFS (unattached).
    - SATA Debian ISO Images link.
  - Install a Debian Guest:
    - File/signature name: "LFSB.ova"/"LFSB.ova.asc" (armor, detach).
    - root passwd: "osp"; hostname: "osp"; new account: "GitHubAccount".
    - install and update more packages.
  - Deliveries: Tarball "`TXT/myW08.tar.bz2.txt`" (signed, asymetric-key, armor) of folder `W08/` (**4 files**).
    1. Blank File: "YourAccount-YourStudentID.txt" (eg. "cbkadal-2006123456.txt").
    2. File: "LFSA.ova" (Blank OVA).
    3. File: "LFSA.ova.asc" (signature, armor, detach).
    4. File: "LFSB.ova.asc" (signature, armor, detach).
- Keep but **DO NOT SUBMIT** file `LFSB.ova`.

Week 08: Check List (Deadline: Monday, 30-Nov-2020).

- Week 08: Linux From Scratch (part 1)(**os08.pdf**).
  1. Week 08 Token: **LINUXFROMSCRATCH**
  2. Read: (OSC10 chapter 05).
  3. Update your Virtual Guest.
  4. Visit `https://os.vlsm.org/GitHubPages/`. Review **Last Week TOP 10 List** and pick at least 3 out of your 10 closest neighbors. See `https://cbkadal.github.io/os211/TXT/myrank.txt`.
  5. Create your **TOP 10 List** of Week 08 (e.g. `https://cbkadal.github.io/os211/W08/`). **Do not use lecture material. Please be more creative!**
  6. Run "`chktoken LINUXFROMSCRATCH`" and write the result into `myW08token.txt`.
  7. Update your log (e.g. `https://cbkadal.github.io/os211/TXT/mylog.txt`).
  8. Update bash script (e.g. `https://cbkadal.github.io/os211/TXT/myscript.sh`).
  9. Make `SHA256SUM` and sign it (detached, armor) as `SHA256SUM.asc`.
  10. Place the assignment result into `W08/` folder and "tar" it into `myW08.tar.bz2`.
  11. GnuPG encrypt (armor) to `myW08.tar.bz2.txt`.

The End

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