

# CSGE602055 Operating Systems

## CSF2600505 Sistem Operasi

### Minggu 05: Virtual Memory

Rahmat M. Samik-Ibrahim

Universitas Indonesia

<http://rms46.vlsm.org/2/207.html>

REV66 7-Sep-2017

|           |                      |  |
|-----------|----------------------|--|
| Minggu 00 | 29 Aug - 05 Sep 2017 | Intro & Review                                       |
| Minggu 01 | 07 Sep - 12 Sep 2017 | IPR, REGEX, & Scripting                              |
| Minggu 02 | 14 Sep - 19 Sep 2017 | Protection, Security, Privacy,<br>& C-language       |
| Minggu 03 | 26 Sep - 30 Sep 2017 | BIOS, Loader, Systemd, & I/O                         |
| Minggu 04 | 03 Okt - 07 Okt 2017 | Addressing, Shared Lib, Pointer<br>& I/O Programming |
| Minggu 05 | 10 Okt - 14 Okt 2017 | Virtual Memory                                       |
| Ming. UTS | 15 Okt - 24 Okt 2017 |  |
| Minggu 06 | 26 Okt - 31 Okt 2017 | Concurrency: Processes & Threads                     |
| Minggu 07 | 02 Nov - 07 Nov 2017 | Synchronization                                      |
| Minggu 08 | 09 Nov - 14 Nov 2017 | Scheduling<br>& Network Sockets Programming          |
| Minggu 09 | 16 Nov - 21 Nov 2017 | File System & Persistent Storage                     |
| Minggu 10 | 23 Nov - 28 Nov 2017 | Special Topic: Blockchain                            |
| Cadangan  | 30 Nov - 09 Des 2017 |  |
| Ming. UAS | 10 Des - 23 Des 2017 |  |

# Week 05: Memory

- 1 Start
- 2 Week 05
- 3 Memory
- 4 Paging
- 5 Translation
- 6 Memory
- 7 Hierarchical
- 8 VM
- 9 Lab
- 10 The End

- Reference: (OSCE2e ch7/8) (UCB 11 12 13) (UDA P3L2) (OLD 06)
- Binding & Linking
  - Address Binding
  - Address Space: Logical & Physical
  - Dynamic & Static Linking
  - MMU: Memory Management Unit
  - Base and Limit Registers
  - Swapping
  - Mobile Systems Problem: no swap
- Memory Allocation
  - Contiguous Allocation
  - Multiple-variable-partition Allocation
  - First, Best, Worst Fit Allocation Strategy
- Fragmentation
  - External
  - Internal
  - Compaction

- Address Space
- Logical/Virtual Address
- Pages
- Page Number
- Page Offset
- Page Table
- PTE: Page Table Entry
- Page Flags: Valid/ Invalid
- TLBs: Translation Look-aside Buffers/ Associative Memory
- Physical Address
- Frames

# Address Translation Scheme

| Address |     | Binary |    |      |    |     |      |     |      |     |  |
|---------|-----|--------|----|------|----|-----|------|-----|------|-----|--|
| DEC     | HEX | OFFSET | PG | OFF  | PG | OFF | PAGE | OFF | PAGE | OFF |  |
| 00      | 00  | 00000  | 0  | 0000 | 00 | 000 | 000  | 00  | 0000 | 0   |  |
| 01      | 01  | 00001  | 0  | 0001 | 00 | 001 | 000  | 01  | 0000 | 1   |  |
| 02      | 02  | 00010  | 0  | 0010 | 00 | 010 | 000  | 10  | 0001 | 0   |  |
| 03      | 03  | 00011  | 0  | 0011 | 00 | 011 | 000  | 11  | 0001 | 1   |  |
| 04      | 04  | 00100  | 0  | 0100 | 00 | 100 | 001  | 00  | 0010 | 0   |  |
| 05      | 05  | 00101  | 0  | 0101 | 00 | 101 | 001  | 01  | 0010 | 1   |  |
| 06      | 06  | 00110  | 0  | 0110 | 00 | 110 | 001  | 10  | 0011 | 0   |  |
| 07      | 07  | 00111  | 0  | 0111 | 00 | 111 | 001  | 11  | 0011 | 1   |  |
| 08      | 08  | 01000  | 0  | 1000 | 01 | 000 | 010  | 00  | 0100 | 0   |  |
| 09      | 09  | 01001  | 0  | 1001 | 01 | 001 | 010  | 01  | 0100 | 1   |  |
| 10      | 0A  | 01010  | 0  | 1010 | 01 | 010 | 010  | 10  | 0101 | 0   |  |
| 11      | 0B  | 01011  | 0  | 1011 | 01 | 011 | 010  | 11  | 0101 | 1   |  |
| 12      | 0C  | 01100  | 0  | 1100 | 01 | 100 | 011  | 00  | 0110 | 0   |  |
| 13      | 0D  | 01101  | 0  | 1101 | 01 | 101 | 011  | 01  | 0110 | 1   |  |
| 14      | 0E  | 01110  | 0  | 1110 | 01 | 110 | 011  | 10  | 0111 | 0   |  |
| 15      | 0F  | 01111  | 0  | 1111 | 01 | 111 | 011  | 11  | 0111 | 1   |  |
| 16      | 10  | 10000  | 1  | 0000 | 10 | 000 | 100  | 00  | 1000 | 0   |  |
| 17      | 11  | 10001  | 1  | 0001 | 10 | 001 | 100  | 01  | 1000 | 1   |  |
| 18      | 12  | 10010  | 1  | 0010 | 10 | 010 | 100  | 10  | 1001 | 0   |  |
| 19      | 13  | 10011  | 1  | 0011 | 10 | 011 | 100  | 11  | 1001 | 1   |  |
| 20      | 14  | 10100  | 1  | 0100 | 10 | 100 | 101  | 00  | 1010 | 0   |  |
| 21      | 15  | 10101  | 1  | 0101 | 10 | 101 | 101  | 01  | 1010 | 1   |  |
| 22      | 16  | 10110  | 1  | 0110 | 10 | 110 | 101  | 10  | 1011 | 0   |  |
| 23      | 17  | 10111  | 1  | 0111 | 10 | 111 | 101  | 11  | 1011 | 1   |  |
| 24      | 18  | 11000  | 1  | 1000 | 11 | 000 | 110  | 00  | 1100 | 0   |  |
| 25      | 19  | 11001  | 1  | 1001 | 11 | 001 | 110  | 01  | 1100 | 1   |  |
| 26      | 1A  | 11010  | 1  | 1010 | 11 | 010 | 110  | 10  | 1101 | 0   |  |
| 27      | 1B  | 11011  | 1  | 1011 | 11 | 011 | 110  | 11  | 1101 | 1   |  |
| 28      | 1C  | 11100  | 1  | 1100 | 11 | 100 | 111  | 00  | 1110 | 0   |  |
| 29      | 1D  | 11101  | 1  | 1101 | 11 | 101 | 111  | 01  | 1110 | 1   |  |
| 30      | 1E  | 11110  | 1  | 1110 | 11 | 110 | 111  | 10  | 1111 | 0   |  |
| 31      | 1F  | 11111  | 1  | 1111 | 11 | 111 | 111  | 11  | 1111 | 1   |  |

# Memory (20 bits)

|       | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | A  | B  | C  | D  | E  | F  |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 00000 | A0 | A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 | A9 | AA | AB | AC | AD | AE | AF |
| 00010 | B0 | B1 | B2 | B3 | B4 | B5 | B6 | B7 | B8 | B9 | BA | BB | BC | BD | BE | BF |
| 00020 | C0 | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | C9 | CA | CB | CC | CD | CE | CF |
| 00030 | D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | DA | DB | DC | DD | DE | DF |
|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| FFFF0 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

# Hierarchical Page Table

- OPT: outer page table (P1)
- PT: page table (P2)
- Offset (D)
- Three-level Paging Scheme
- Hashed Page Tables
- Inverted Page Table



- Demand Paging
- COW
- Page Replacement
- Frame Allocation
- Kernel

- Lab

- `putchar(char)`
- `getpid()`
- `getppid()`
- `sprintf(char*, const char*)`
- `fflush(NULL)`
- MSIZE1 (10k) MSIZE2 (20k) MSIZE3 (50k) MSIZE4 (100k)  
MSIZE5 (1M) MSIZE6 (10M) MSIZE1
- `top`
  - PID (Process Id), PPID (Parent PID), %MEM (Memory), VIRT (Virtual Image KiB), RES (Residen Size KiB), SHR (Shared Memory KiB), SWAP (Swapped Size KiB), CODE (Code Size KiB), DATA (Data+Stack KiB), USED (Res+Swap Size KiB).
  - Save: `~/.toprc`
  - `top -b -n 1 -pYOUR_PID`
- `malloc(size_t)`
- `free(void*)`
- `system(const char*)`

# The End

- This is the end of the presentation.