* Shorena K. Anzhilov / CS5600 / Northeastern University

* Spring 2025 / March 2025

* Week 8 - Assignment / Page Tables and Page Faulting

Part 1.

Question 1: Page tables

Virtual Page #	Directory Index (i)	Table Index (i)	Physical Block #	Writeable (w)
0080C	008	00C	00003	0 (RO)
0080D	008	00D	00004	0 (RO)
0080E	008	00E	00005	0 (RO)
0100A	010	00A	00006	1 (RW)
0100B	010	00B	00007	1 (RW)
0100C	010	00C	00008	1 (RW)

Mapping Representation (Text-Based Arrow Format)

- **Page Directory (00000)**
  - **Entry 008 → Page Table 00001**
  - **Entry 010 → Page Table 00002**
- **Page Table 00001 (Read-Only Pages)**
  - **Index 00C → Physical Page 00003 (RO)**
  - **Index 00D → Physical Page 00004 (RO)**
  - **Index 00E → Physical Page 00005 (RO)**
- **Page Table 00002 (Read/Write Pages)**
  - **Index 00A → Physical Page 00006 (RW)**
  - **Index 00B → Physical Page 00007 (RW)**
  - **Index 00C → Physical Page 00008 (RW)**

The Second part:

Part 2.

Question 2: Page Faulting

Initial State:

- □ **Stack Pointer (SP) = 0C001,000** (before decrementing by 4 for PUSH)
- □ **Program Counter (PC) = 00000,FFC**
- □ **Page Directory (`00000`) exists, but all entries are marked non-present**
- □ **Available physical pages:** `00001` to `00009`

Instruction Execution Trace:

1. Instruction: `PUSH #10` (at `00000,FFC`)

- **Instruction Fetch:**
  - Fetch instruction from `00000,FFC`
  - **Fault:** Page `00000` is **not in memory**
  - **Allocate physical page `00001` for virtual page `00000`**
  - **Read block `0` from `/bin/program` into `00001`**
  - **Updadte Page Directory → Set `00000[0]` = `00001`**
  - **Return from fault.**
  - **Retry instruction fetch (`00000,FFC`) → Success.**
- **Execution of `PUSH #10`**
  - **Stack Pointer decreme → Page Fault**
  - **Fault:** Page `0C000` is **not in memory**
  - **Allocate physical page `00002` for page `0C000`**
  - **Updadte Page Set → `00000[C]` = `00002`**
  - **Return from fault.**
  - **Retry instruction (`PUSH #10`) → Success.**

---

2. Instruction: `CALL 2,000` (at `00001,000`)

- **Instruction Fetch (`00001,000`)**
  - **Fault:** Page `00001` is **not in memory**
  - **Allocate physical page `00003` for Virtual page `00001`**
  - **Read block `1` from `/bin/program` into `00003`**

    - **Set 00000[1] = 00003**
    - **Return from fault.**
    - **Retry fetch from 00001,000 → Success.**
- **Execution of CALL 2,000**
    - **Stack Pointer decrements: 0C000,FFC → 0C000,FF8**
    - **Store return address (00001,004) at 0C000,FF8 → Success.**
    - **Jump to 00002,000**

---

3. Instruction: MOV EAX → *(10,000) (at 00002,000)

- **Instruction Fetch (00002,000)**
    - **Fault:** Page 00002 is **not in memory**
    - **Allocate physical page 00004 for page 00002**
    - **Read block 2 from /bin/program into 00004**
    - **Set 00000[2] = 00004**
    - **Return from fault.**
    - **Retry fetch from 00002,000 → Success.**
- **Execution of MOV EAX → *(10,000)**
    - **Store at 00010,000**
    - **Fault:** Page 00010 is **not in memory**
    - **Allocate physical page 00005 for page 00010**
    - **Set 00000[10] = 00005**
    - **Return from fault.**
    - **Retry store at 00010,000 → Success.**

---

4. Instruction: HALT (at 00002,004)

- □ **Instruction Fetch (00002,004) → Success.**
- □ **Program Terminates.**

Final Summary of Steps:

- **Fetch PUSH #10 at 00000,FFC → Page Fault**
    - Allocate 00001, load from /bin/program, set 00000[0] = 00001
    - Retry fetch → Success.
- **Attempt PUSH #10**

    - Store at 0C000,FFC → Page Fault.
    - Allocate 00002, set 00000[C] = 00002
    - Retry store → Success.
- **Fetch CALL 2,000 at 00001,000 → Page Fault**
    - Allocate 00003, load from /bin/program, set 00000[1] = 00003
    - Retry fetch → Success.
- **Attempt CALL 2,000**
    - Store at 0C000,FF8 → Success.
    - Jump to 00002,000.
- **Fetch MOV EAX → *(10,000) at 00002,000 → Page Fault**
    - Allocate 00004, load from /bin/program, set 00000[2] = 00004
    - Retry fetch → Success.
- **Attempt MOV EAX → *(10,000)**
    - Store at 00010,000 → Page Fault.
    - Allocate 00005, set 00000[10] = 00005
    - Retry store → Success.
- **Fetch HALT at 00002,004 → Success.**
    - **Program Terminates.**

Step	**Action**	**Outcome**
1	Fetch PUSH #10 at 00000,FFC	**Page Fault**
	Allocate 00001, load from /bin/program, set 00000[0] = 00001	**Resolved**
	Retry fetch	**Success**
	Attempt PUSH #10	**Success**
	Store at 0C000,FFC	**Page Fault**
	Allocate 00002, set 00000[C] = 00002	**Resolved**
	Retry store	**Success**
2	Fetch CALL 2,000 at 00001,000	**Page Fault**
	Allocate 00003, load from /bin/program, set 00000[1] = 00003	**Resolved**
	Retry fetch	**Success**
	Attempt CALL 2,000	**Success**
	Store at 0C000,FF8	**Success**

	Jump to `00002,000`	**Success**
3	Fetch `MOV EAX → *(10,000)` at `00002,000`	**Page Fault**
	Allocate `00004`, load from `/bin/program`, set `00000[2] = 00004`	**Resolved**
	Retry fetch	**Success**
	Attempt `MOV EAX → *(10,000)`	**Success**
	Store at `00010,000`	**Page Fault**
	Allocate `00005`, set `00000[10] = 00005`	**Resolved**
	Retry store	**Success**
4	Fetch `HALT` at `00002,004`	**Success**
	Program Terminates	**Completed**