

# Question 1: Page tables

breakdown virtual memory regions

0080C000 -> pg# = 0080C offset = 000

pg# 0080C -> 0000 0000 10 | 00 0000 1100 -> 002 | 00C

(first 10-bit is index in level\_1 table, last 10-bit is index in level\_2 table)

Thus,

0080EFFF -> pg# = 00800 offset = FFF

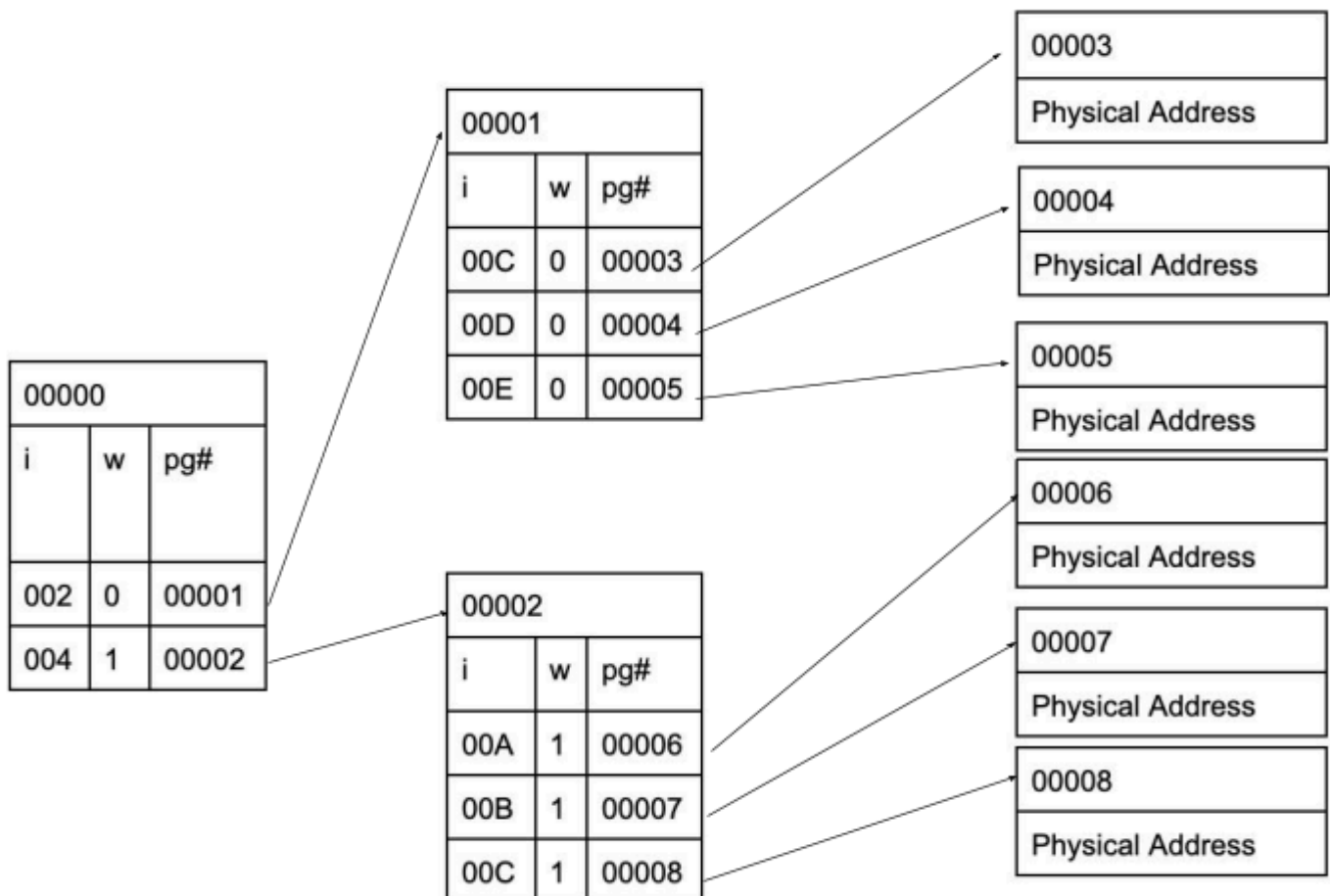
pg# 0080E -> 0000 0000 10 | 00 0000 1110 -> 002 | 00E

0100A000 -> pg# = 0100A offset = 000

pg# 0100A -> 0000 0001 00 | 00 0000 1010 -> 004 | 00A

0100CFFF -> pg# = 0100C offset = FFF

pg# 0100C -> 0000 0001 00 | 00 0000 1100 -> 004 | 00C



## Question 2: Page Faulting

### 00000,FFC PUSH #10

#### Fetch First Instruction (PUSH #10)

- Instruction fetch(00000,FFC)
- Fault(00000,FFC, Instruction PUSH #10)
- Page allocate = 00001
- 00000[000].page = 00001
- Return\_from\_fault
- Retry Instruction Fetch(00000,FFC)
- Fault(00000,FFC, Instruction PUSH #10)
- Page allocate = 00002
- 00001[000].page = 00002
- Load code page from /bin/program
  - read\_block(/bin/program, 0, 00002)
- Return\_from\_fault
- Retry Instruction Fetch(00000,FFC)
- Instruction successfully fetched

#### Execute PUSH #10

- Attempt(PUSH #10)
- Stack Pointer decrements: SP = 0C000,FFC
- Store(0C000,FFC)
- Fault (0C000,FFC, data store)
- Allocate a page for stack: page\_allocate = 00003
- 00000[030].page = 00003
- Return from fault
- Retry store(0C000,FFC)
- Fault (0C000,FFC, data store)
- Allocate a page for stack: page\_allocate = 00004
- 00003[000].page = 00004
- Return from fault
- Retry store(0C000,FFC)
- Success

### 00001,000 CALL 2,000

#### Fetch Instruction (CALL 2,000)

- Instruction Fetch(00001,000)
- Fault (00001,000, Instruction CALL 2,000)
  - 00000[000].page -> 00001 is already allocated
  - 00001[001].page is not allocated
- Allocate a page for code continuation: page\_allocate = 00005
- 00000[001].page = 00005
- Load page from /bin/program
  - read\_block(/bin/program, 1, 00005)
- Return from fault
- Retry Instruction Fetch(00001,000)

- Instruction successfully fetched

### Execute CALL 2,000

- Attempt(CALL 2,000)
- Stack Pointer decrements: SP = 0C000,FF8
- Store return address (00001,004) to SP
  - Store(0C000,FF8)
- Success
- PC = 00002,000

## 00002,000 MOV EAX → \*(10,000)

### Fetch Instruction (MOV EAX → \*(10,000))

- Instruction Fetch(00002,000)
- Fault (00002,000, instruction fetch)
  - 00000[000].page -> 00001 is already allocated
  - 00001[002].page is not allocated
- Allocate a page for code: page\_allocate = 00006
- 00001[002].page = 00006
- Load code page from /bin/program
  - read\_block(/bin/program, 2, 00006)
- Return from fault
- Retry Instruction Fetch(00002,000)
- Instruction successfully fetched

### Execute MOV EAX → \*(10,000)

- Attempt(MOV EAX → \*(10,000))
- Store(00010,000)
- Fault (00010,000, data store)
  - 00000[000].page -> 00001 is already allocated
  - 00001[010].page is not allocated
- Allocate a page for data: page\_allocate = 00007
- 00001[010].page = 00005
- Return from fault
- Retry store(00010,000)
- Success

## 00002,004 HALT

### Fetch Instruction (HALT)

- Instruction Fetch(00002,004)
  - 00000[000].page -> 00001 is already allocated
  - 00001[002].page -> 00006 is already allocated
- Success

### Execute (HALT)

- Attempt(HALT)
- Success
- Program Stops