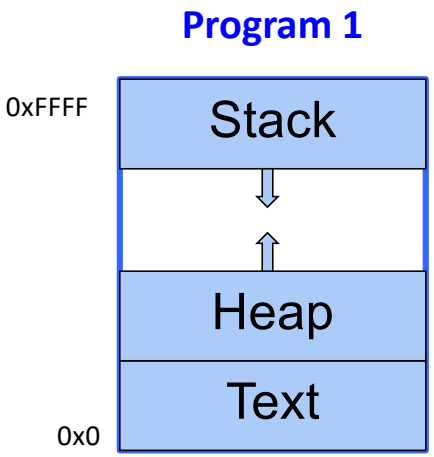
Today

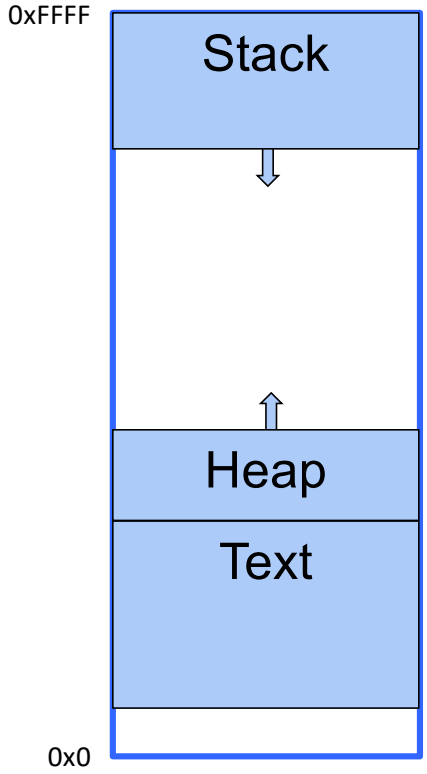
- Roadmap
  - Page Tables: How the OS maps virtual addresses to physical addresses
- Learning Outcomes
  - Define:
    - Page Table
    - Page Table Entry (PTE)
    - Segfault
    - Page Fault
  - Explain what each field in a PTE means
- Reading
  - 9.5

Recall how we represent Address Spaces

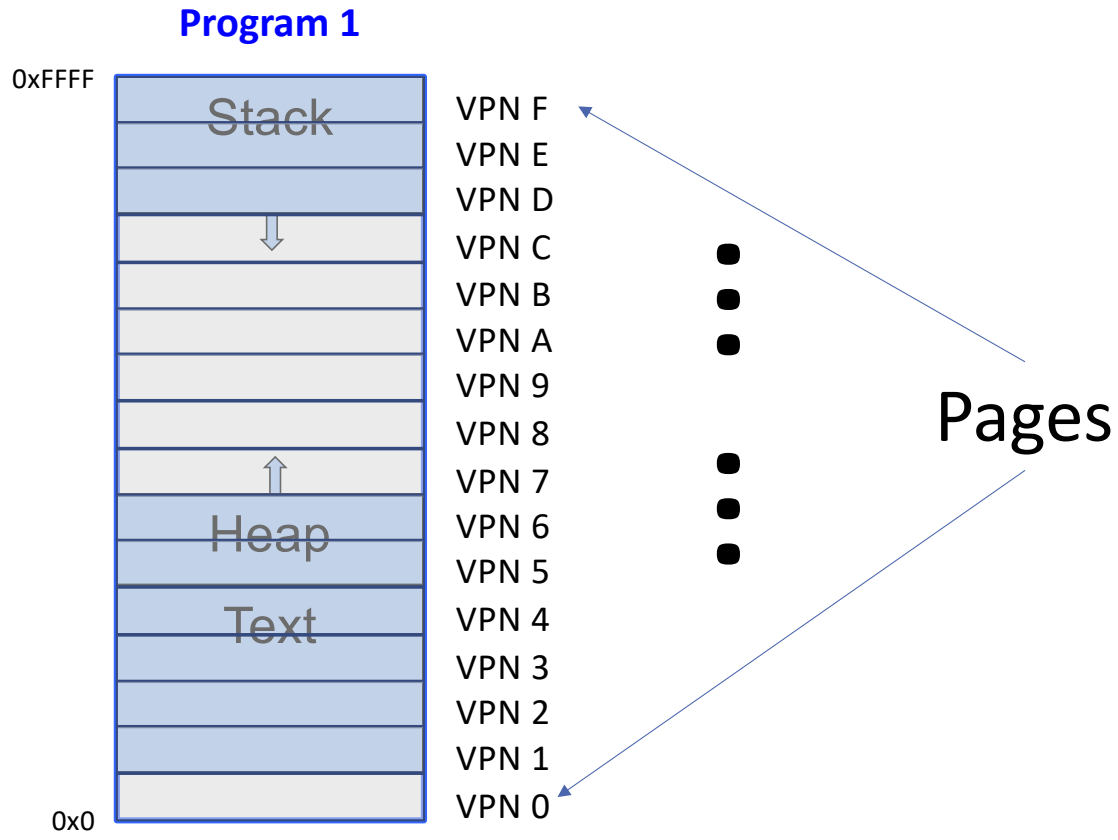


Back to Address Spaces

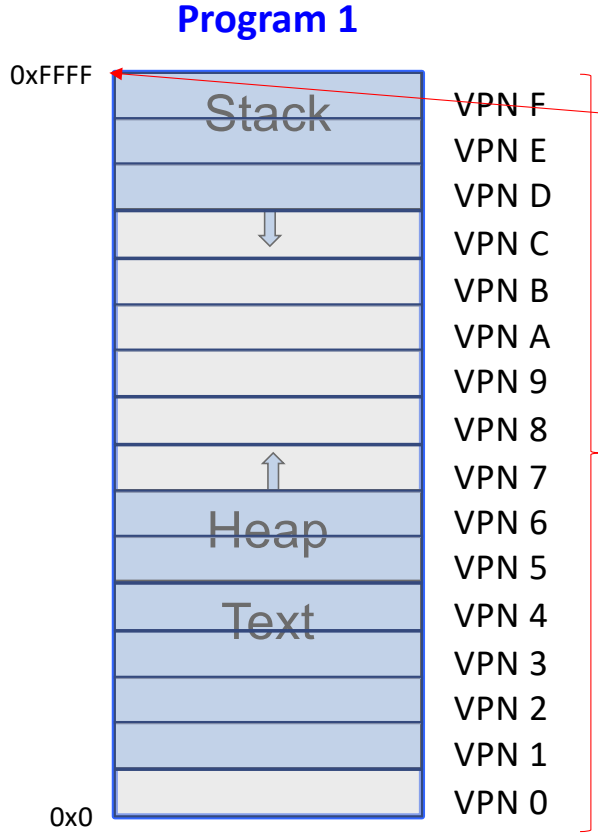
Program 1



Back to Address Spaces



Back to Address Spaces

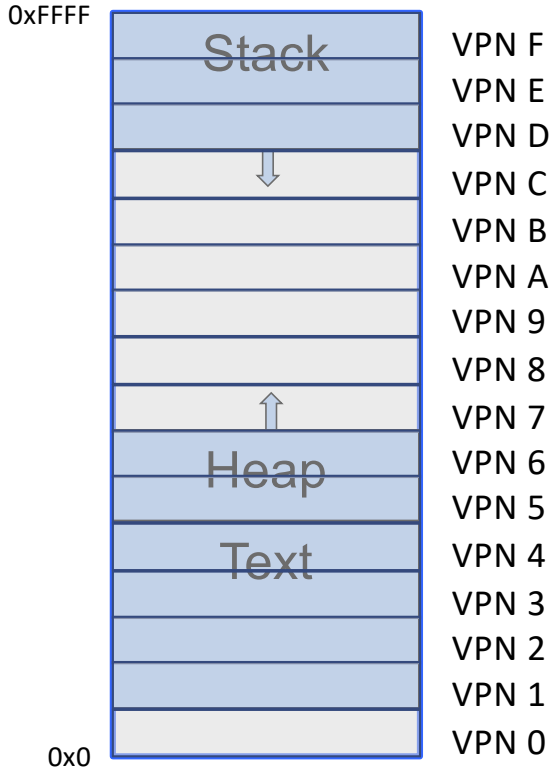


Notice:

- The address space is 16 bits (4 hex digits)
- There are 16 pages (0-F)
- Therefore, this must be a 4 KB page system.

Mappings: Recall

Program 1



(VA, access, privilege) => (PA/fault)

Triple consisting of a:

- Virtual address (VA)
- Access (read, write, execute)
- Privilege (user, supervisor)

Physical address
OR
Fault (turns control over to the operating system

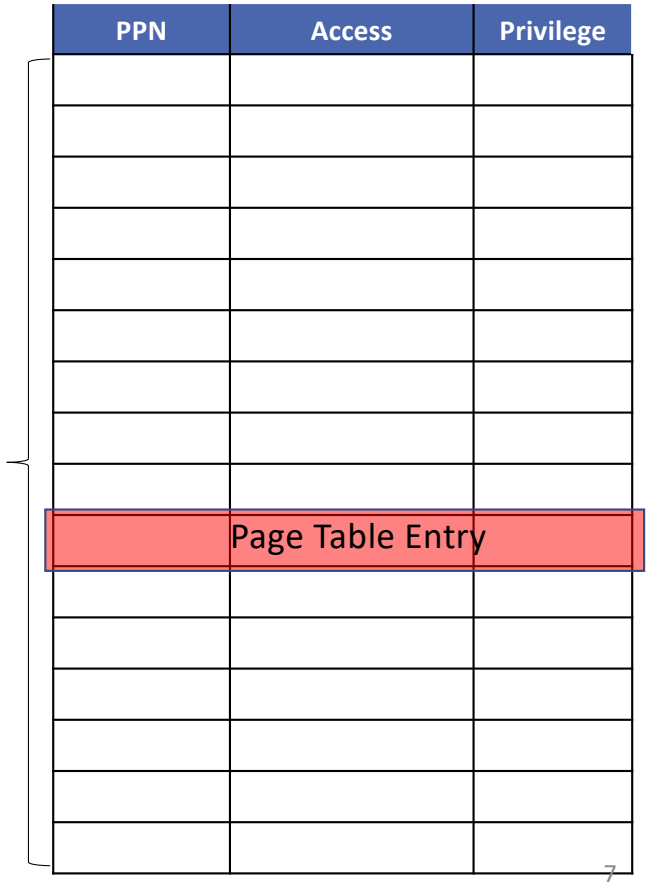
And what would we call that fault?

An exception!

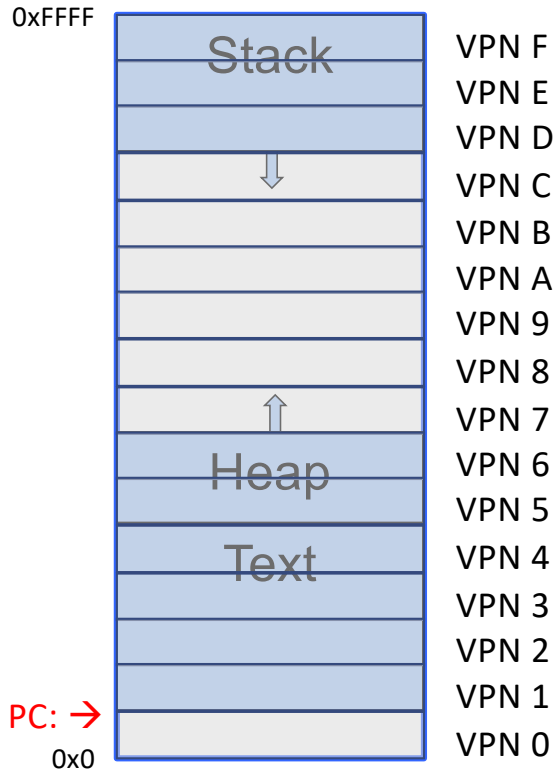
Page Tables: Mapping data structure

- The TLB is simply a cache of mappings.
- The Page Table is the data structure that holds all the mappings.

Indexed by virtual page number (VPN)



Page Tables: A pile of PTEs



Page table

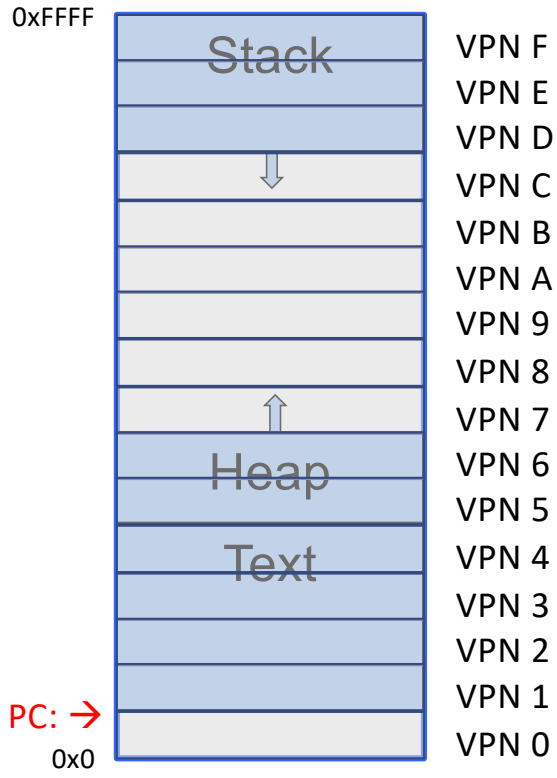
PTE

indices

	PPN	Access	Privilege
0	Invalid		
1	0x0000A	Read, Execute	U
2	0x1100B	Read, Execute	U
3	0x98765	Read, Execute	U
4	0xCAFE0	Read, Execute	U
5	0xFACE1	Read, Write	U
6	0xC0FFF	Read, Write	U
7	Invalid		
8	Invalid		
9	Invalid		
A	Invalid		
B	Invalid		
C	Invalid		
D	0x50505	Read, Write	U
E	0x12345	Read, Write	U
F	0x24680	Read, Write	U

Address Translation

Program 1



Assume we are running in user mode

```
xor %rax, %rax
mrmovq 0xFEED(%rax), %rbx
```

What virtual address are we accessing?

0xFEED

What is the VPN to translate?

0xF

What is the corresponding PTE?

0x24680/RW/U

Do we have the proper permissions?

Yes

What is the PA for the data?

0x24680EED

	PPN	Access	Privilege
0	Invalid		
1	0x0000A	Read, Execute	U
2	0x1100B	Read, Execute	U
3	0x98765	Read, Execute	U
4	0xCAFE0	Read, Execute	U
5	0xFACE1	Read, Write	U
6	0xC0FFF	Read, Write	U
7	Invalid		
8	Invalid		
9	Invalid		
A	Invalid		
B	Invalid		
C	Invalid		
D	0x50505	Read, Write	U
E	0x12345	Read, Write	U
F	0x24680	Read, Write	U

Page Table Entries: PTEs

- A PTE can be in one of three states
  1. Invalid: There is no mapping

     Exception: Segfault (usually kill the process)
  2. Valid and Memory-Resident: Permissions dictate if the access is allowed

     Execution proceeds as normal
  3. Valid but not Memory-Resident: Permissions still dictate if the access is allowed. While the VPN is valid, the page we want to access is not yet in memory; the OS must make it memory resident.
     - Could require reading the page from disk
     - Could require creating a page full of 0's

     Exception: Page fault (read the page from disk and then proceed as normal)

Page Tables: Summary

- The page table is the collection of mappings that describe an address space.
- The page table contains a set of page table entries (PTEs).
- A PTE can be in one of three states:
  - invalid -- physical page number and permissions should be ignored
  - valid and present -- contains physical page number, access and mode info
  - valid and not present -- location of physical page on disk, access and mode info