

SSE3044 Operating Systems
prof. Jinkyu Jeong

Project 2. Virtual memory

2018.5.2 (Wed.)

TAs

이규선(gyusun.lee@cs.skku.edu) /

안민우(minwoo.ahn@cs.skku.edu)

Project plan

- Total 4 projects

- 1) Process management

- 2) Virtual memory

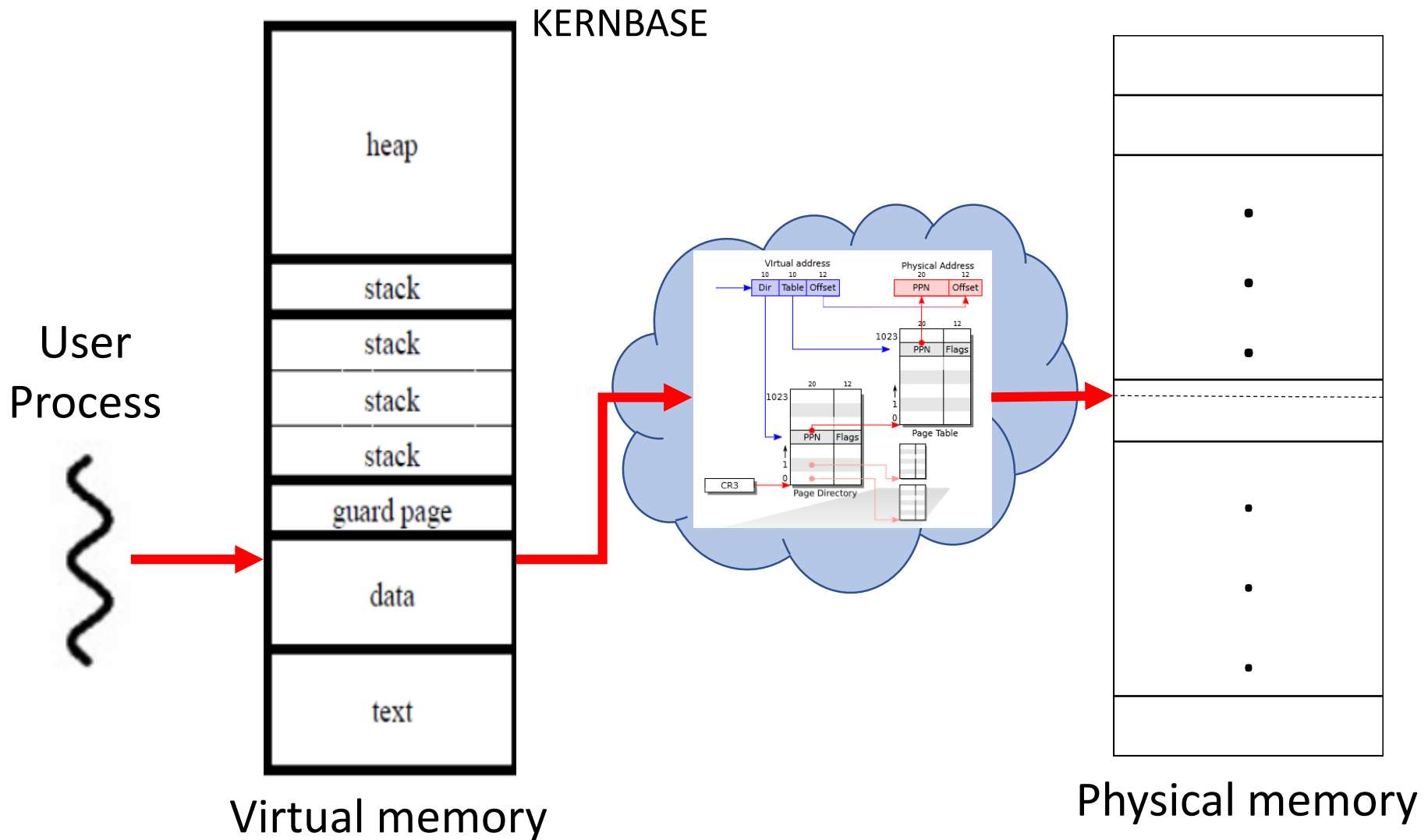
- Stack growth

- CoW(Copy-on-write) (Due: 5/2 ~ 5/15)

- 3) Synchronization

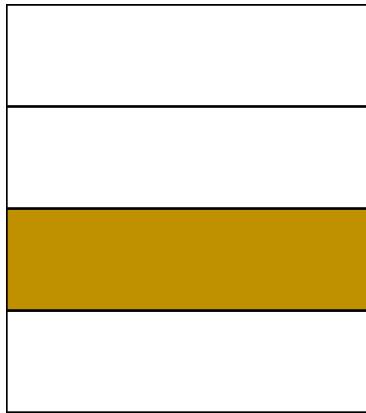
- 4) File system

(Remind) Address translation

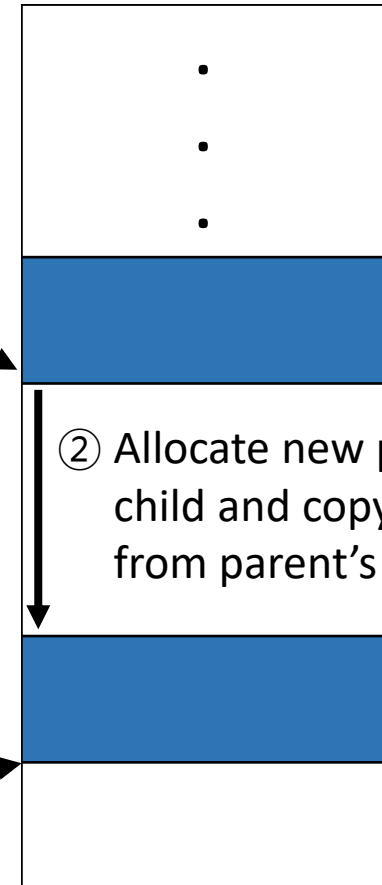
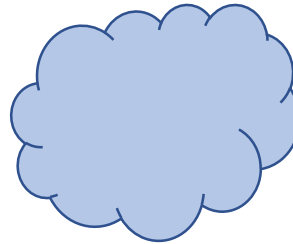


Implementation of fork() in xv6

Parent process's vm

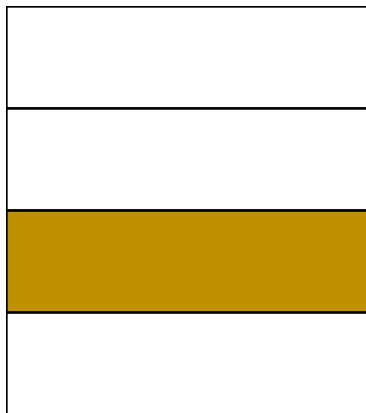


① Translate parent process's virtual address from 0, page by page

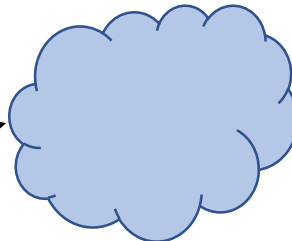


② Allocate new page for child and copy memory from parent's page

Child process's vm



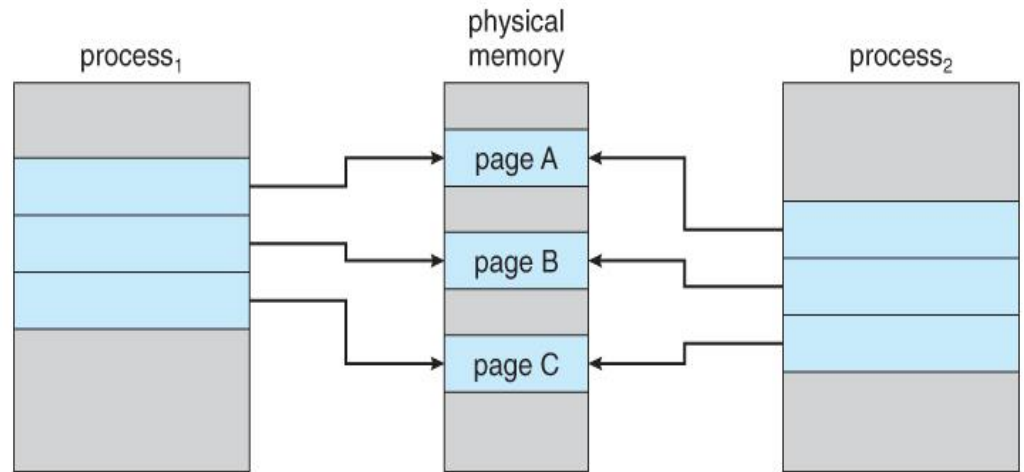
③ Map newly allocated page to child process's page table



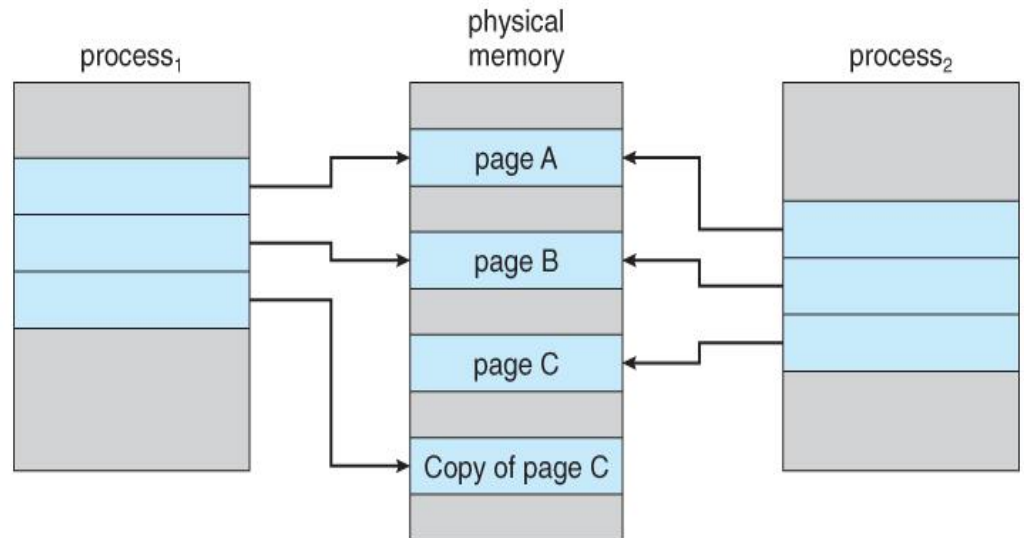
Physical memory

What is Copy-on-Write (CoW)?

After fork() system call
&
No modify on pages



process 1 modifies page C



Project2_2. Copy-on-Write

- ① Use original xv6 code as a base
- ② Add new system call “numfree()” which returns number of free page frame in physical memory
- ③ When process forks,
 - ① Create shared mapping to the same page frames in physical page
 - ② Shared pages are read-only (not writable)
- ④ When data is written to shared pages,
 - ① Page fault is generated
 - ② Should consider faulted read-only page was originally read-only
 - ③ xv6 allocates new page frame in physical memory, and copy memory of original before write
- ⑤ Reference counter for physical pages is needed
- ⑥ `lcr3(V2P(pde_t)) -> flush page table`

Submission

- Compress your code as YourStudentID-2_2.tar.gz
- Send your file to “minwoo.ahn@csi.skku.edu”
- **PLEASE DO NOT COPY**
 - **YOU WILL GET -100 POINTS OF YOUR PROJECT IF YOU COPIED**
- Due date: 5/15(Sun.), 23:59:59 PM

Grade policy

- If you failed to pass “oral test”, you will get 0

Questions

- If you have “**any**” questions, please email to TA
- You can also visit Semiconductor Building #400509
 - Please email TA before visiting