

CIS 452 - Operating Systems Concepts

Nathan Bowman

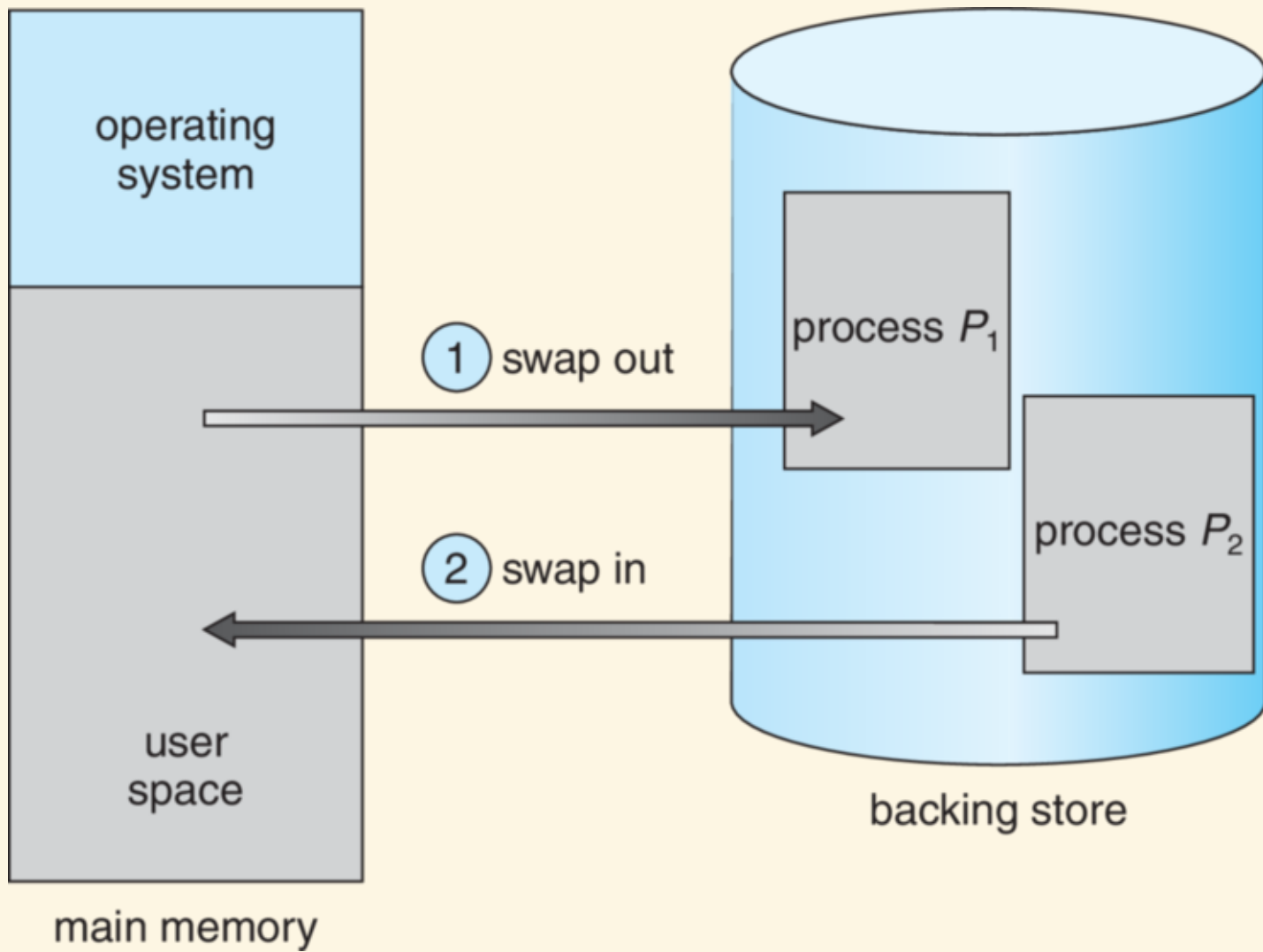
Images taken from Silberschatz book

Swapping

We will discuss various ways of managing memory in
OS

Simple way to allow multiprogramming is **swapping**

Processes are temporarily moved out of memory and
into a backing store such as a hard disk



Swapping is simple way of handling our memory management problems

If processes are swapped at every context switch, each process will own all of memory (except for OS memory)

Processes would always know where they were being loaded, so absolute addressing would be possible

Major downside is that context switch becomes very expensive

Assume user process is 100 MB

Speed of backing store is 50 MB/second

Transfer of process from disk to memory takes 2 seconds (2000 milliseconds)

Outgoing process (of same size) must be transferred to disk, so additional 2 seconds required

Time depends heavily on size of processes

Must also be sure that process is *completely* idle before swapping out

What if process is waiting for I/O?

I/O may asynchronously access user memory as I/O buffer

If different process is put into that memory, I/O may perform incorrect read or may overwrite memory incorrectly

To avoid this, must either

- never swap process with pending I/O, or
- use only OS buffers for I/O operations

Second solution results in increased overhead due to **double buffering** -- data is transferred into OS buffer, then OS transfers into user buffer when appropriate

Performance of system based entirely on swapping
would be unacceptable

Modified version of swapping used only when system is
low on memory