

OS 4/23

Reagan Shirk

April 23, 2020

Chapter 10: Virtual Memory

- Virtual memory abstracts physical memory to be a **large** array of storage
 - benefits both the user and the system
 - system: able to optimize and take care of memory locations/partitions, no worries about user error
 - separates the logical memory that the user/programmer developer needs from the physical memory
 - there are shared libraries in virtual memory
 - shared memory across processes
 - * means you have to take care of who is requesting what and be aware of what happens on/in these memory... things? I missed the last word
- Referencing the picture at the bottom of the diagram, how do we choose what memory to bring in from the backing storage and what to remove?

fork vs vfork

- **vfork** is faster
 - it doesn't copy all of the memory

Copy on Write

- Don't create a new page until a write operation occurs

Page Replacement Algorithms

- Find location in storage
- Find a free frame
 - If found, use it
 - If no free frame, use a page-replacement algorithm
 - * Identify a victim frame
 - Write the contents of the victim frame to storage
 - * Update the tables
- Read in the new page and continue
- Different algorithms:
 - Optimal (need all future page requests)
 - * The fastest, obvi
 - Least recently used
 - * Not as fast as optimal but still pretty good
 - FIFO
 - * The worst
 - Clock Algorithm
 - * Eh not too great

Kernel Memory allocation

- Buddy system
 - Cuts memory by a factor of 2 buddies until the memory request is fit
 - Deallocates recursively

Supporting Pictures and stuff

