ITP 30002 Operating System

Complete Virtual Memory System

OSTEP Chapter 23

Shin Hong

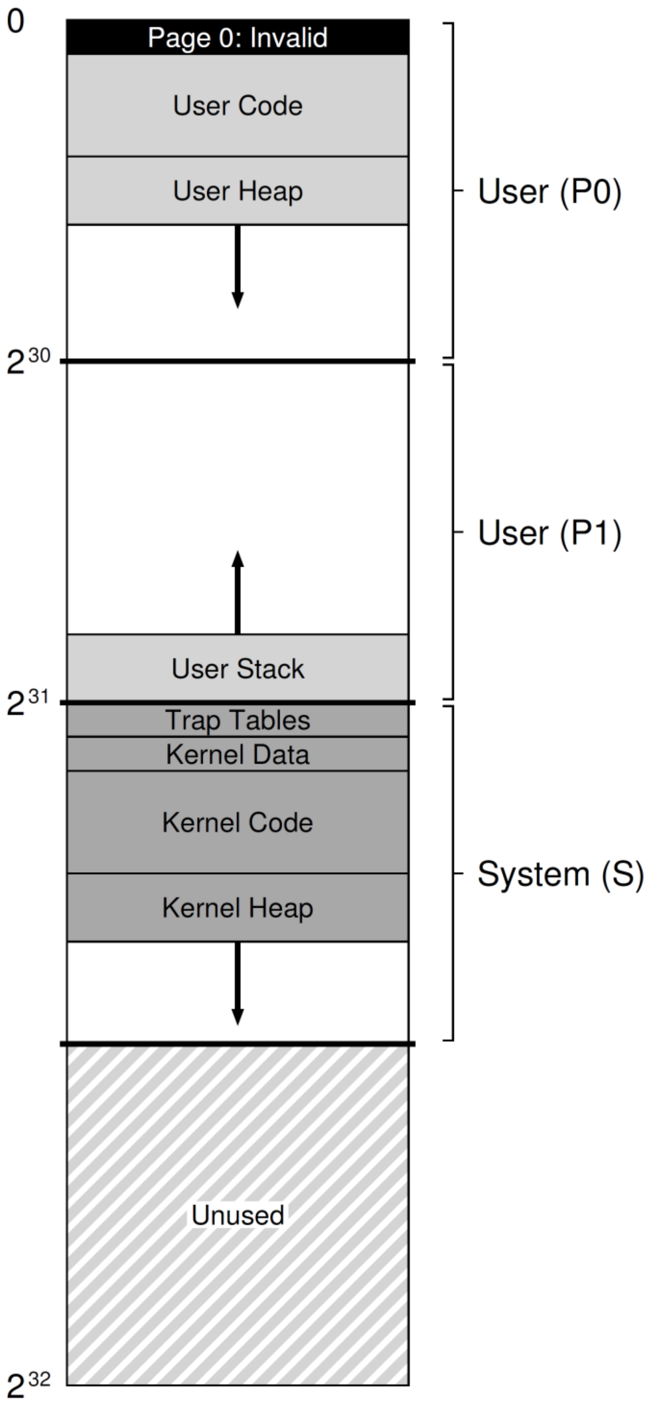
VAX/VMS Virtual Memory

- VAX/VMS is OS for VAX-11, a minicomputer architecture invented in 1970’s by Digital Equipment Corporation (DEC)

- MMU of VAX-11
  - 32-bit virtual address space per process with 512-byte pages
    - 23-bit VPN and 9-bit offset
  - top two bits of VPN indicates whether an address is the first half or the second half of process space, or system space
    - two page tables per process
    - bounds register is used for holding the number of pages in the space
    - keep page tables in the kernel space
    - the kernel can swap pages of the page tables out to disk

VAX/VMS Address Space Layout

- Page 0 is not used and marked inaccessible
- kernel address space is a part of each user address space
  - kernel is like a library with a protection



VAX/VAM Page Replacement

- Unfortunately, VAX architectures do not have reference bit in a page table entry

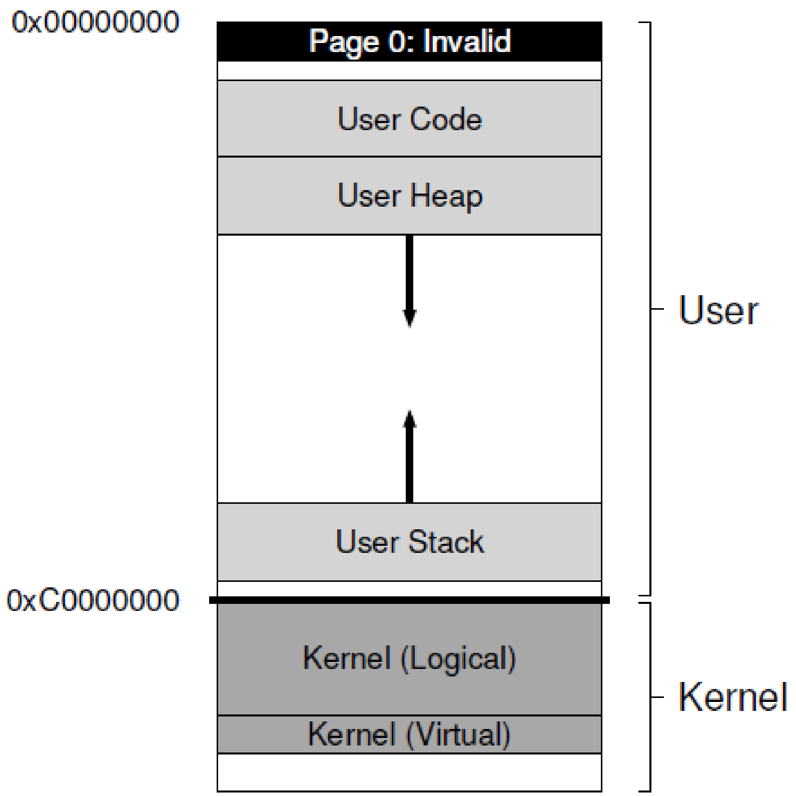
- Segmented FIFO replacement policy
  - each process has a max number of pages to keep them in memory (resident set size; RSS) and keeps the pages in a FIFO
  - clock replacement algorithm using modified bit

Lazy Optimization for Responsiveness

- demand zeroing
  - at a process creation, assign a page table entry and mark the page inaccessible
  - pursue zeroing when the process accesses the page for the first time (via trap)

- copy-on-write
  - after a fork, a page table entry of a child process points to the corresponding page of the parent process
  - duplicate a page from the parent process when a child process updates the page

Linux VM System for x86

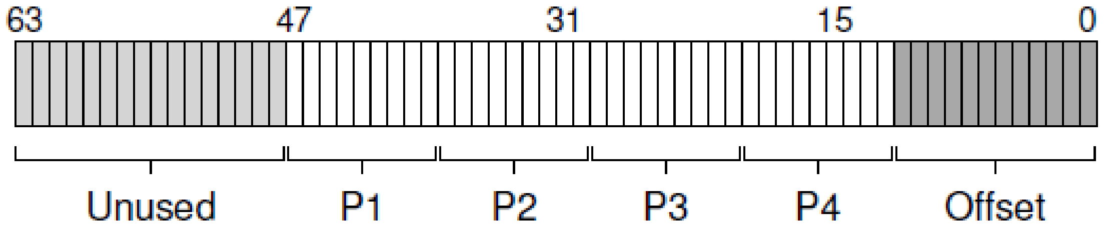


- Address space is separated into **user portion** and **kernel portion**
  - for 32-bit x86, three quarters are for user portion and one quarter for kernel portion
- Kernel virtual address space comprises of
  - kernel logical addresses
    - normal virtual addresses for kernel space
    - directly mapped to the first portions of physical memory
    - continuous in physical memory addresses
    - cannot be swapped
  - kernel virtual addresses
    - may be swapped and non-continuous in physical addresses
    - much flexible to hold large data in kernel space

Page Table

- x86 provides a hardware-managed, multi-level, per-process page tables

- 64-bit x86 uses a four-level page table that uses 36 bits for VPN



- Linux provides transparent large-size page supports
  - Benefit of having large-size pages (e.g., 2-MB page, 1-GB page)
    - less TLB miss
    - shorter path of TLB miss handling

Page Replacement Policy – 2Q

- based on LRU
- has two lists of pages, inactive list and active list
  - to resolve the limitation of LRU on treating large file accesses
- policies
  - a page is placed on inactive list first when it is first brough in
  - a page is promoted to active list when the page is re-referenced
  - a pages in inactive list is first selected as a victim for a page replacement

Security Support

- Memory errors (or other simple errors) may become targets for security attacks and cause malicious users to take control of the system
  - ex. buffer overflow

```
int some_function(char *input) {
  char dest_buffer[100];
  strcpy(dest_buffer, input); // oops, unbounded copy!
}
```

- Defense mechanism
  - No-execute bit (NX bit) of a page entry: prevent execution of any code found within certain pages
  - Address space layout randomization (ASLR): randomize placement of code, stack and heap in order to make it impossible to inject code to fixed locations