Compsys Recap 10/01/24

Understanding, races and Virtual address translation
Af: TA Jóhann Utne

Agenda

- OS in general
- Races
- Virtual memory and virtual address translation
- Exam exercises

OS in general

- process vs program
  - program is the “dead” code
  - process is the “live” instance running the program
- Kernel
  - “Aways resident code that services request from the hardware and manages processes”
  - Unprivileged, must make calls to kernel to switch to privileged state (interrupts)
  - The kernel handles: hardware, system memory, File I/0 and context switching

Processes, Threads and races

- Processes
  - duplicate of parent, but with own address space (changes are not reflected)
  - Child processes must be reaped, adopted but init process if termination of parent. Processes that never terminates are “zombie children”
- Threads
  - run in same address as the calling process (changes are reflected)
  - have own thread context: ID, SP, PC, general purpose registers, condition codes
  - can access “critical memory” must be handled with semaphores (mutexes) and / or condition variables
- Races
  - code depend on order of execution

Fork example from exam:

```c
int main () {
  if (fork() == 0) {
    printf("1");
    if (fork() == 0) {
      printf("2");
    }
  } else {
    pid_t pid = fork();
    if (waitpid(pid, NULL, 0) > 0) {
      if (fork() == 0) {
        printf("3");
      } else {
        printf("4");
      }
    }
    printf("5");
  }
}
```

Virtual addresses

- Why is virtual memory useful?:
  - caching
  - memory management
  - memory protection
- Components of virtual address:
  - VPN, VPO, Tag og Index
  - TLB
  - PPN and PPO (physcial adresses)
  - Page table (fully associative)

Translation of virtual adresses

- split the address in its components VPO, VPN, and then split the VPN in index and tag
- bits of the VPO = log2(pageSize)
- bits of VPN = rest
- bits of set = log2(#sets) bits of VPN
- bits of Tag = rest of vpn
- Good idea to mark this
- Translate hex-adresses to bit addresses (each digit corresponds to 4 bits)

Translation cont. (algorithm for translating)

- read description to find page size and #sets
- translate the hex to binary
- Find VPN, index and set
- look up in TLB with index and set
  - if the tag exists AND the valid bit is set it is a hit, otherwise a miss.
- if no hit in TLB look in page table , with VPN
  - if VPN exits AND the valid bit is set hit, otherwise page fault and a new page must be brought from memory
- If hit, copy ppn and ppo = vpo

Exam set:

Exam-set 2022- 23

if time

Re-Exam-set 2022-23