Recitation 2: Intro to OS + Kernel & user space processes & threads

Mohammad H. Mofrad
University of Pittsburgh

hasanzadeh@cs.pitt.edu
What is a computing resource?

- **Central Processing Unit (CPU):** Fetch + Decode + Execute operations
- **RAM (Random Access Memory):** Acts as temporary storage and Virtual Memory implementation
- **Hard Disk Drive (HDD):** Uses magnetic storage (SCSI, SATA, USB)
- **Network Interface Card (NIC):** LAN or Wireless adaptors that connects computers to networks
- **Graphic card:** Feeds images to display. Video Graphics Array (VGA) is one of the display hardware standards.
- **Input/output (I/O) devices:** Device drivers for I/O operations
Who’s in charge of resources?

- **Management**: Keep track of resources
- **Abstraction**: Masking OS internals using User Interfaces (UIs), e.g. Graphical User Interface (GUI) or Command Line Interface (CLI)
- **Sharing**: Allocating resources (e.g. CPU, memory, I/O devices) between different processes
- **Time**: Time sharing for multi user operating systems
- **Security**: Protect users from other users and attackers using mechanisms like authentication, encryption, isolation, and Access Control List (ACL)
What are different types of Operating Systems (OSs)?

• **Proprietary** e.g. Apple iOS, Microsoft windows and etc.

• **Non proprietary** e.g. Linux, Google Chromium OS, Android and etc.

• **Network OSs** e.g. Cisco IOS

https://en.wikipedia.org/wiki/List_of_Linux_distributions
Kernel and User spaces

- **User space** refers to the code outside of OS where apps are running
- **Kernel space** refers to the code for OS kernel and device drivers
- What this separation brings to us?
  - Memory protection (0x00000000)
  - Hardware protection (STD IO)
- **User mode** enables users to get access to installed applications and available libraries (e.g. C standard library)
- **Kernel mode** enables kernel to get access to Scheduler, InterProcess Communication (IPC), Virtual Memory (VM), Virtual File Systems (VFSs), Network Interfaces, etc.

Privilege rings

https://en.wikipedia.org/wiki/Operating_system
What’s the missing link between U&K spaces?

printf(“Hello, World!\n”);  
Display(Hello, World!)

A context switch is the switching of the CPU from one process to another.
System Call (Syscall)

• The system call is the fundamental interface **between** an application and the **Linux kernel**.

• System calls are generally not invoked directly, but rather via wrapper functions in glibc (or perhaps some other library).

• Heretofore, there are ~400 syscalls in Linux

• Some famous syscalls:
  • chdir, chmod, chown
  • bind
  • open, close, exit
  • getpid
  • ioctl, signal

http://man7.org/linux/man-pages/man2/syscalls.2.html
Microkernel VS Monolithic Kernel

• User VS Kernel access
• Context Switch overhead
• Inter VS Intra communication cost
• Access time
• Security via isolation
• BUGs
• # <codes>
• Flexibility
Tanenbaum VS Torvalds

Microkernel

Kernel

Servers

IPC

Software

Monolithic kernel

Kernel

Software

https://en.wikipedia.org/wiki/Tanenbaum%E2%80%93Torvalds_debate
Processes and Threads

• A **process** is a executing instance of a program
• **fork()** in Linux
  • SySCALL
  • Parent and child relationship between parent and forked process
  • A scheduling unit can be seen as a process
  • A word processor may have a single process
  • Each process has a separate memory space

• A **thread** is a stream of execution
• **pthread** library in c (**pthread_create()**, **pthread_join()**)
  • C standard library
  • Shared memory
  • The grammar check plugin of the word processor is a thread
Process states

- Process states: New, running, waiting, ready, terminated
- Process Control Block (PCB)
  - Process state
  - Program Counter (PC)
  - CPU registers
  - CPU scheduling information
  - Memory management info.
  - Accounting information
  - I/O status
Process in memory

Local variables, pointes

Dynamically allocated space e.g. malloc, or coalloc

Global and static variables, and constant data types, and pointers

Constant data types

0x0000

0xFFFF

0x0000

0xFFFF

stack

heap

data

code
Single thread & Multiple threads processes

![Diagram showing single and multiple threads processes across user space and system space.](image)
User Thread VS Kernel Thread

- Scheduling
- Thread switch
- Context Switch
- Bookkeeping
- Non blocking IO
- Blocking IO
- Speed