Introduction to Operating System

Lecturer： 周喆、张凯

Overview

- Textbook
  - **Operating Systems: Three Easy Pieces**
  - https://pages.cs.wisc.edu/~remzi/OSTEP
  - Andrea C. Arpaci-Dusseau & Remzi H. Arpaci-Dusseau
    Wisconsin-Madison
- Labs
  - We will **NOT** follow XV6,But much simpler labs based on Linux Kernel
  - We will release manual soon.
  - Just prepare a notebook with full battery

Marking Schemes

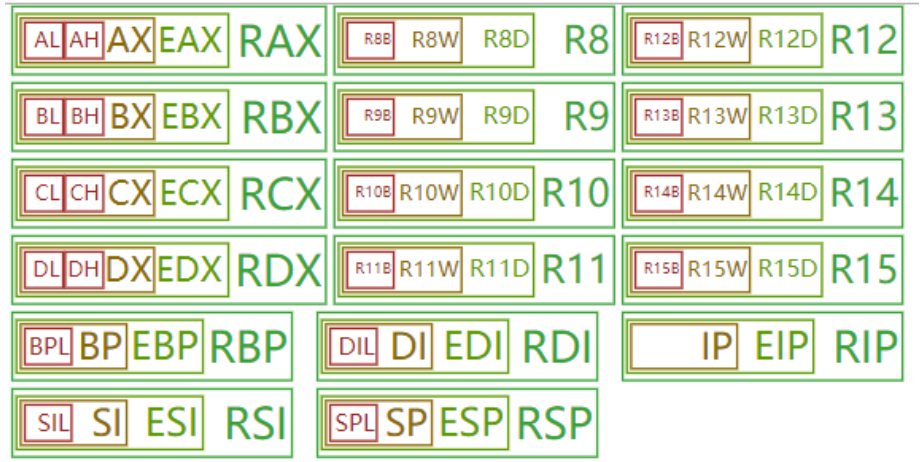
- Final Examination: 60%
- Labs: 40%
  - TA will interview students for EVERY lab.
  - Do not cheat! Do not copy and paste codes.

Prerequisite

- ICS
  - malloc, synchronization
  - Processes, signal, I/O
- C Programming
  - function pointers
  - multi-thread programming
- x86 ISA assembly
  - inline assembly
  - registers, opcodes, memory indexing
  - calling convention

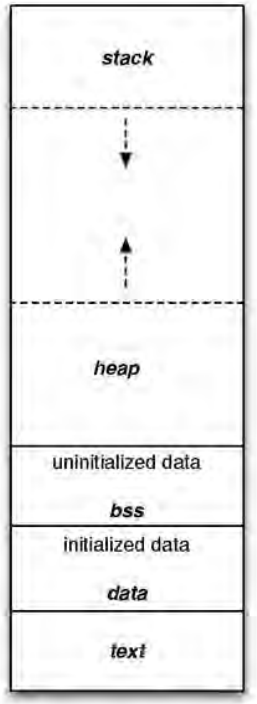
Review – Assembly Basics

- Regs: ABCD: Accumulator, Base, Counter, Data; SI, DI: Src, Dst.
- Memory Indexing
  - mov RAX, QWORD PTR FS:[RBX + RCX*4+ 0x8]
- Opcodes: mov add inc sub push pop
- Calling Convention
  - Volatile Registers: ACD…
  - Non-volatile: B, DI, SI…



Review – malloc()

- brk() vs mmap()
  - brk increase the heap
  - mmap finds a empty location between heap and stack
- 128K is the threshold
  - smaller memory is served by heap
    - buddy algorithm to manage fragments
  - larger by mmap
    - let OS to manage
  - Why?



Review – inline assembly

```
// x86-64 Linux
#include <asm/unistd.h>      // compile without -m32 for 64 bit call numbers
// #define __NR_write 1
ssize_t my_write(int fd, const void *buf, size_t size)
{
    ssize_t ret;
    asm volatile
    (
        "syscall"
        : "=a" (ret)
        //                 EDI      RSI       RDX
        : "0"(__NR_write), "D"(fd), "S"(buf), "d"(size)
        : "rcx", "r11", "memory"
    );
    return ret;
}
```

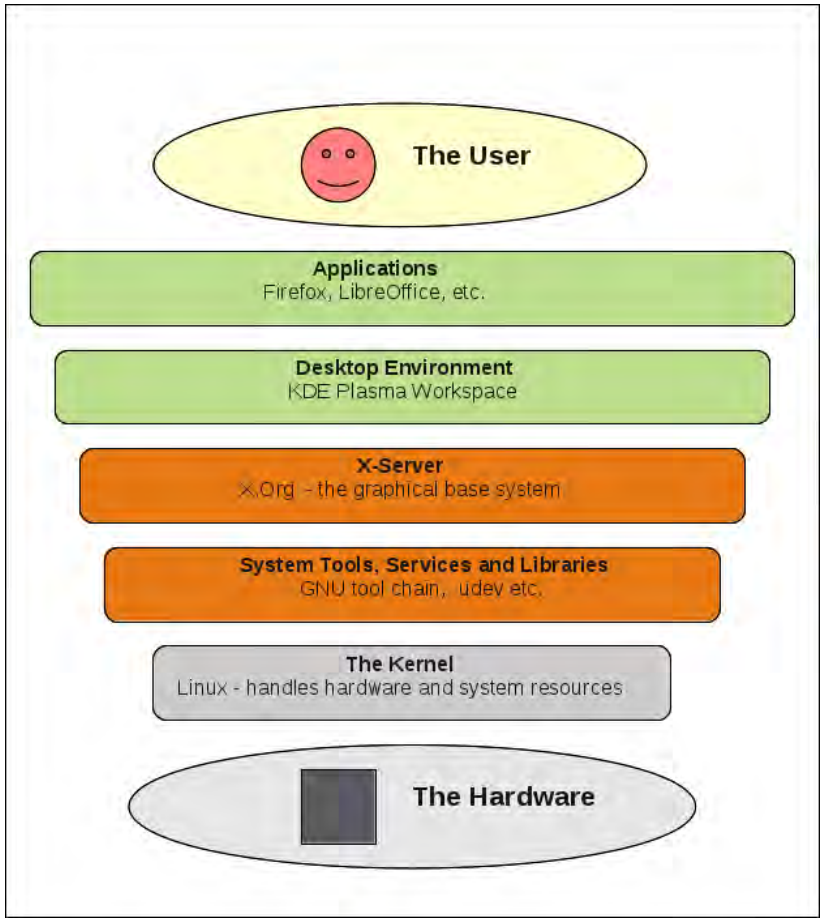
What is OS?


Nathan
My Documents
My Recent Documents
My Pictures
My Music
My Computer
Control Panel
Printers and Faxes
Help and Support
Search
Run...
All Programs
Log Off
Turn Off Computer
start
Take a tour of Windows XP
To learn about the exciting new features in XP now, click here. To take the tour later, click All Programs on the Start menu, and then click Accessories.


100%
La Fuerte
Shakira & Bizarr
Focus
Entrance
Door
Locked
Bedroom
Speaker
Not Playing


Linux™

What is OS?



Daemons and runtime libraries
also matter!!!

What does OS provides?

- In a word: **Abstraction** – Provide standard library for resources
- What is a resource?
  - Anything valuable (e.g., CPU, memory, disk)
- What abstraction does modern OS typically provide for each resource?
  - CPU: process and/or thread
  - Memory: address space
  - Disk: files
- Advantages of OS providing abstraction?
  - Allow applications to reuse common facilities
  - Make different devices look the same
  - Provide higher-level or more useful functionality
- Challenges
  - What are the correct abstractions?
  - How much of hardware should be exposed?

What does OS provides?

- Behind Abstraction: Resource management – Share resources well
- Advantages of OS providing resource management?
  - Protect applications from one another
  - Provide efficient access to resources (cost, time, energy)
  - Provide fair access to resources
- Challenges
  - What are the correct mechanisms?
  - What are the correct policies?

OS Organization: Three Pieces

Virtualization

Make each application believe it has each resource to itself

Concurrency

Events are occurring simultaneously and may interact with one another

Persistency

Lifetime of information is longer than lifetime of any one process

A Brief History of OS

- The most important system in history: UNIX
  - Based on advanced designs, such as Multics from MIT
  - Written in C, Open Source
  - MacOS, BSD, SunOS (Sun), AIX (IBM)
- Composability vs monolithic design.
  - Write programs that **do one thing and do it well**.
  - Write programs to work together.
  - Write programs to handle text streams
  - POSIX standard
- Legal problems -> Linux