Welcome to CS 449!

Instructor: Anatoli Shein

Introduction
and
class rules

Introduction

Anatoli Shein

- From Belarus
- PhD Student
- Research Interests
- Advisors
- Background

Introduction (cont.)

Students

- Name
- Freshman/Sophomore/Junior/Senior
- Major

Some Rules

- Mobile phones – silent mode
- Ask the Instructor
- Food & Drinks in the classroom
- Ask the TA

The syllabus

http://people.cs.pitt.edu/~aus/cs449.html

1. What is systems programming?

- The act of writing **systems code**, which constructs the environment that allows applications to execute in it.
- **Systems code** is managing hardware (i.e. CPU, main memory, peripheral devices) and provides common services to applications such as file storage and network service.

2. Why is learning system programming important?

- Industry needs people with knowledge in building systems
- Get deeper understanding about the environment in which applications are executing

3. Systems vs Applications Programming

Applications	Systems
Deals with abstractions, objects	Deals with registers, memory locations
Solves human problems	Controls and manages systems
Device/platform independence	Directly targets specific devices/systems
Java, C#, Python, garbage collection	C, x86 assembly, manual memory alloc.
Big fat runtime systems	Resource constraint environment

Both can be efficient if done properly, and it is important to learn both.

4. What is a process?

- Process is an instance of a program, which is currently executing.
- Program → Compilation → Executable (binary) → Process

5. How do we know if the process is doing useful work?

- It produces viewable output
- Importance of INTERACTION between process and system
- Java vs C

6. Layers of the system

Layer 3: Application, User-level libraries

Layer 2: Device-independent OS code, File System, Network Stack

Layer 1: Device-dependent OS code, Device Drivers

Layer 0: Hardware

7. Communication between layers

Example of **system call** (sending a network packet).

- **Layer 3**: The application writes a message to a network socket by calling a socket API in glibc (the C runtime library). The glibc library issues a write system call on the socket.
- **Layer 2**: The message traverses the multiple layers of the TCP/IP protocol stack to form a packet and then the packet gets sent to the underlying Ethernet device driver.
- **Layer 1**: The Ethernet device driver talks to the underlying Ethernet network card to write the packet into the internal buffer of the card.
- **Layer 0**: The Ethernet card sends along the packet in its buffer through the physical LAN line.

[**System call**: layers 3 to 0 vs **signal**: layers 0 to 3]

8. Why learn C?

- Vast majority of system code is written in C
- Executes on native machine (not virtual)

9. Pros and Cons of C

Pros	Cons
Can operate in resource-constrained environments	No concept of object oriented programming
Is very efficient and has little runtime overhead	No language safety features
Allows for direct control over memory access and control flow	No automatic garbage collection of memory
Lets the programmer write parts of the program directly in assembly language	

10. The Hello World Program

in **C**

```
#include <stdio.h>

int main(int argc, char *argv[])
{
    printf(“Hello world!\n”);
    return 0;
}
```

In **Java**

```
class HelloWorldApp
{
    public static void main(String[] args)
    {
        System.out.println("Hello world!");
    }
}
```

11. Data Types

Integers	**Reals**	**Text**
short int long long long	float double long double	char

- Signed vs. unsigned
- Pointers
- The sizeof() operator