

COM 410 – Slides 1

Tour of computer systems
and
the C programming language

Vocabulary

Compiler

Source Code

Assembly language

Object code

Executable Code

Busses

I/O devices

Main Memory

Cache Memory

Central Processing Unit (CPU)

Operating System

Concurrency

Context switching

Threads

Register

Memory hierarchy

Virtual memory

Process

File

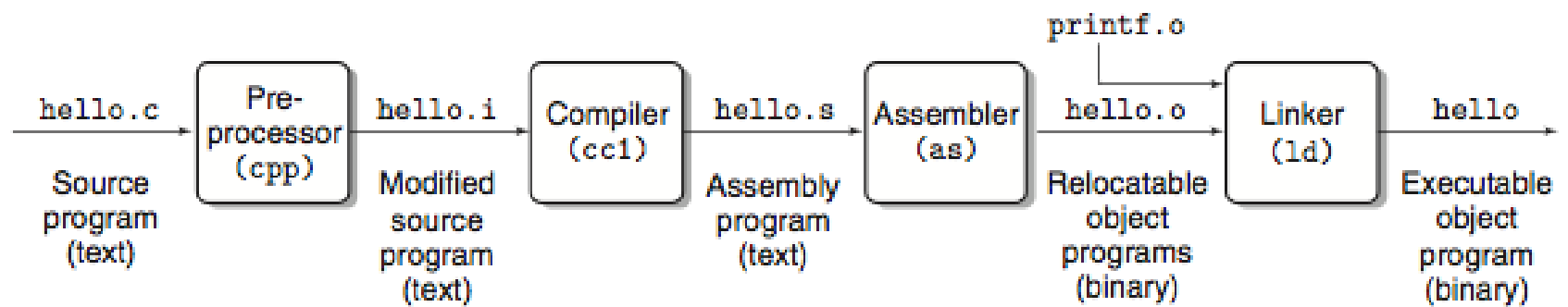
Network

```
1  #include <stdio.h>
2
3  int main()
4  {
5      printf("hello, world\n");
6      return 0;
7  }
```

```
linux> gcc -o hello hello.c
```

```
1  #include <stdio.h>
2
3  int main()
4  {
5      printf("hello, world\n");
6      return 0;
7  }
```

This “source file” (hello.c) is stored as a bunch of 0s and 1s representing characters – to be translated by a compiler into machine code

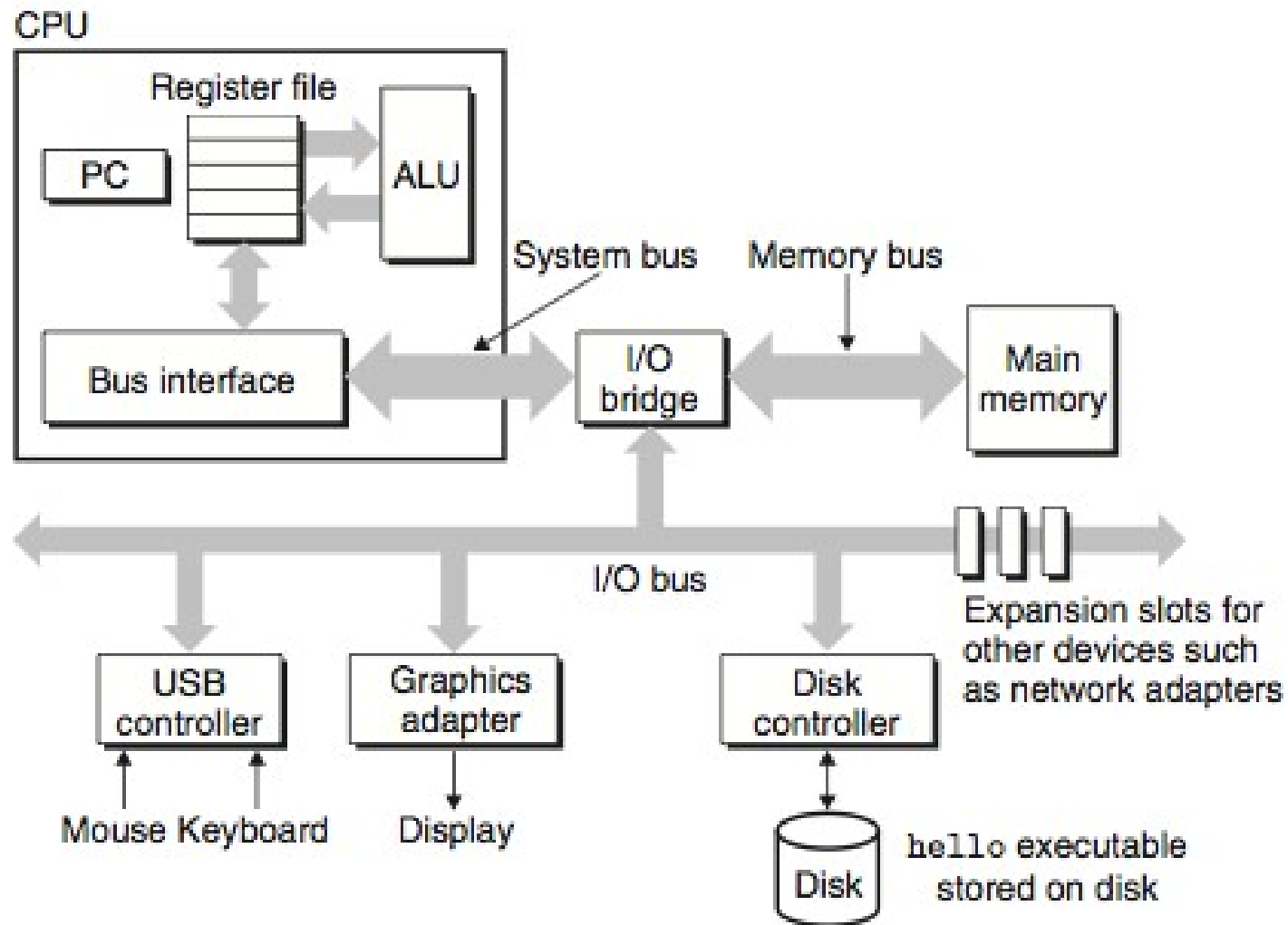


```
1  main:
2      subq    $8, %rsp
3      movl    $.LC0, %edi
4      call    puts
5      movl    $0, %eax
6      addq    $8, %rsp
7      ret
```

Assembly language version of 'hello.c'

Why understand how compilation works:

- Is a switch statement more efficient than a sequence of if-else statements?
- How much overhead is used in a function call?
- is a while loop more efficient than a for loop?
- Are loops faster if using a local variable, or an argument passed by reference?
- Understanding linking errors in big programs
- Avoiding security holes, like buffer overflow vulnerabilities!



hardware organization of a typical system

Buses

Carry 'words' of data between components, e.g. 32 or 64 bits (4 or 8 bytes)

IO devices

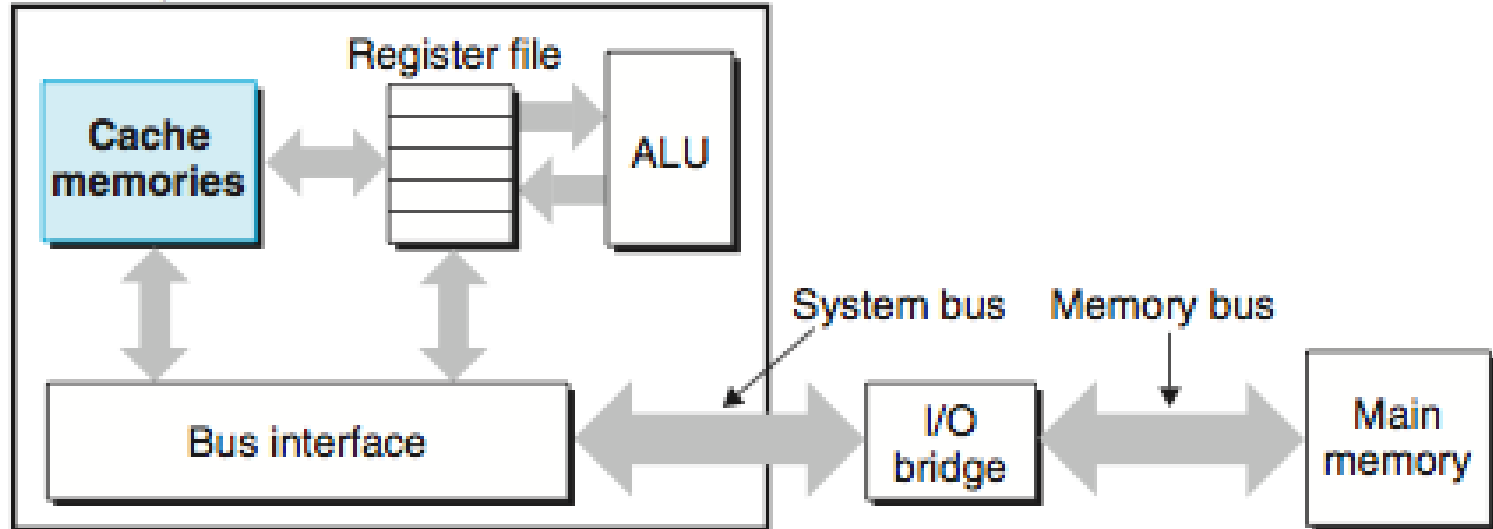
Keyboard, mouse, monitor, disk drive

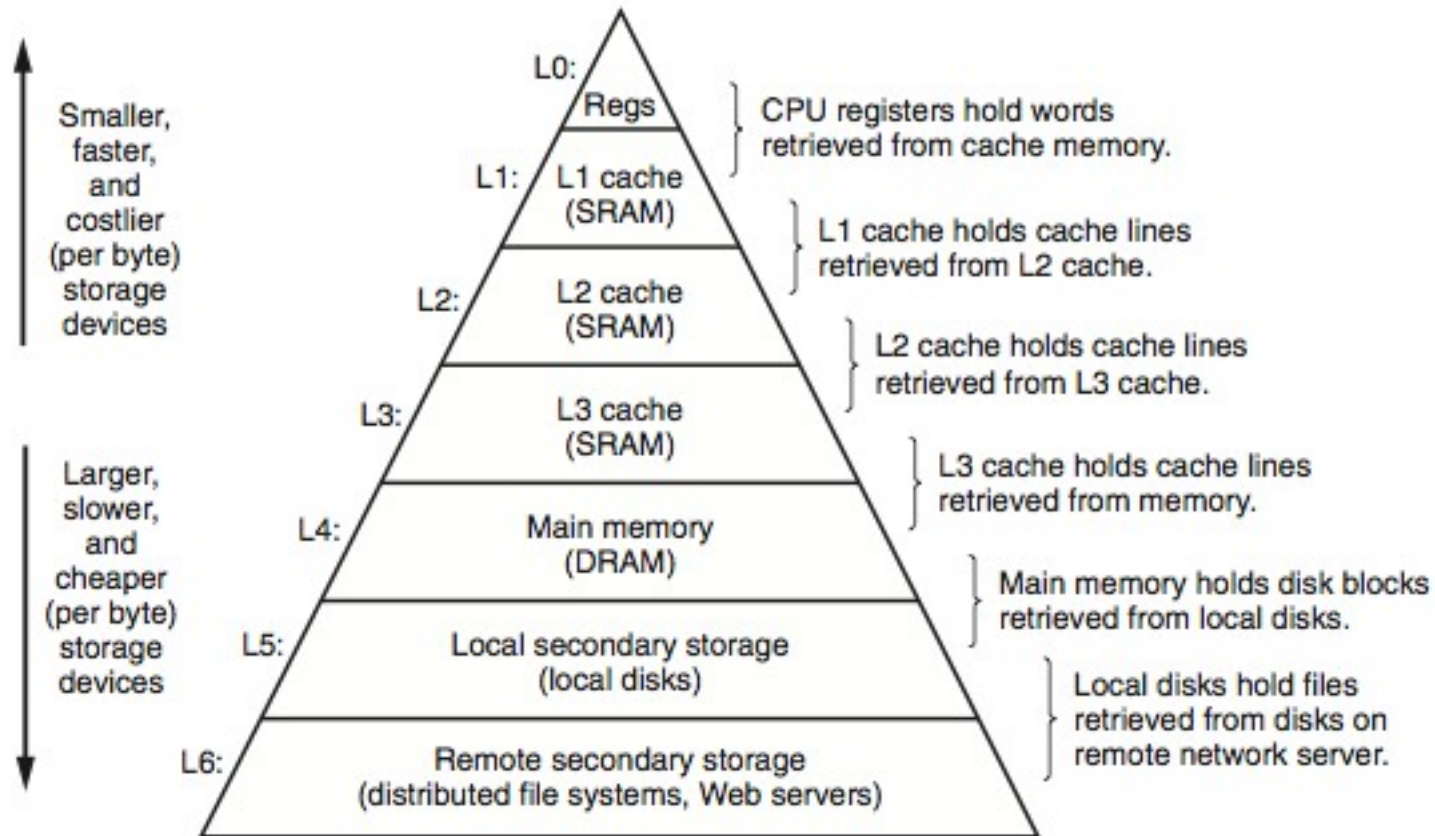
Main Memory

DRAM

Processor (CPU, other types of processors)

CPU chip





types of cache memory
("memory hierarchy")

Operating System:

Gives the illusion that your program is the only one running, but your processes (your program) is being run *concurrently* with other processes, sharing the CPU(s), memory, and IO devices.
Called “*context switching*”

The operating system’s job is to protect hardware from misuse by runaway applications, and managing wildly different hardware devices from computer to computer

Threads:

In modern systems, a process (program) can consist of multiple execution units, called “threads,” each thread running in the context of the process and sharing the same code and global data. It is easier to share data between multiple threads than multiple processes – way to make programs run faster

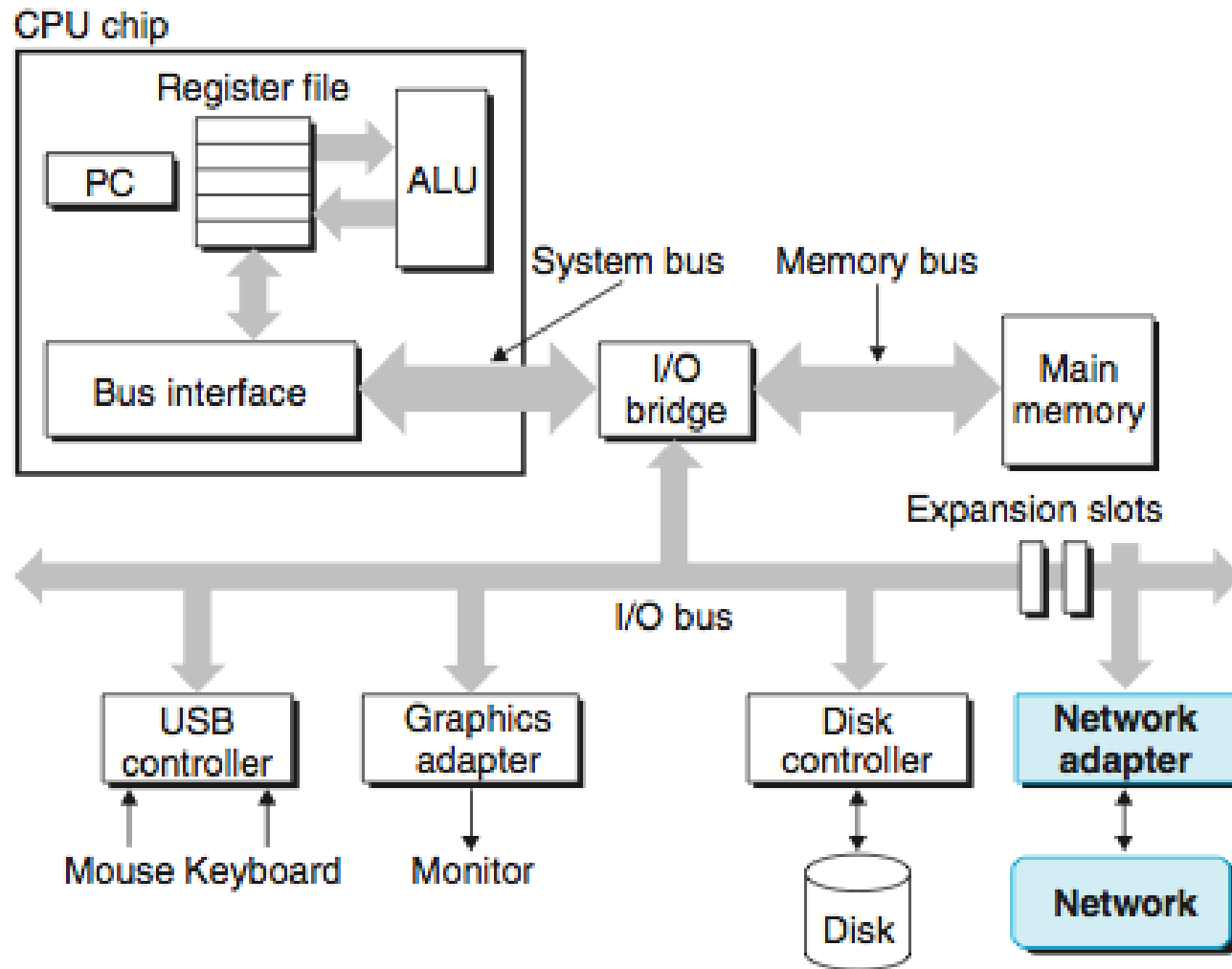
Virtual Memory:

Provides processes with the illusion that it has exclusive use of main memory (DRAM)

The “heap” and the “stack.” Pointers to pointers... Ugh!

Files

Sequence of bytes, stored on disk. Every I/O device including disks, keyboards, displays, and even networks, is modeled as a file.



Systems communicate with other systems using networks (a network can be viewed as just another I/O device).

C Programming Language

- Loops (for, while)
- Conditionals (if, else, switch)
- Recursion (calling a function from itself)
- Command line arguments to main()
- Make file
- Formatted output (printf)

- Pointers!!!
- Pass by reference, pass by value