Compilation and Assembly

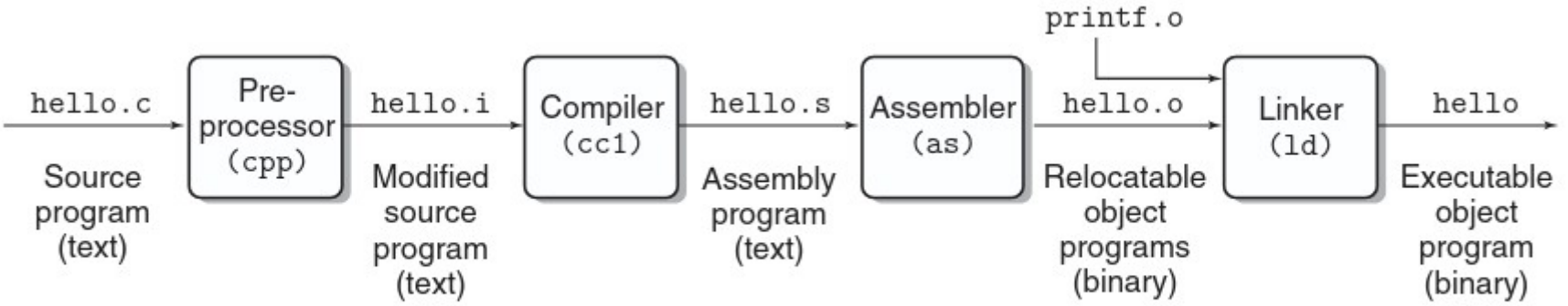


Figure 1.3 **The compilation system.**

Preprocessing

- ‘#’ Directives
- Include Files
- gcc -E -o hello.i hello

```c
#include <stdio.h>

int main () {
  printf("HELLORD\n");
  return 0;
}
```

```
wc -l hello.i
751 hello.i
head hello.i
# 0 "hello.c"
# 0 "<built-in>"
# 0 "<command-line>"
# 1 "/usr/include/stdc-predef.h" 1 3 4
# 0 "<command-line>" 2
# 1 "hello.c"
# 1 "/usr/include/stdio.h" 1 3 4
# 27 "/usr/include/stdio.h" 3 4
# 1 "/usr/include/bits/libc-header-start.h" 1 3 4
# 33 "/usr/include/bits/libc-header-start.h" 3 4
```

Compilation (Assembly)

- Preprocessor output is targeted to hardware
- Defaults to host hardware
- Note structure of ASM
- Last human readable step

```
gcc -S hello.i -o hello.s
cat hello.s
        .file   "hello.c"
        .text
        .section        .rodata
.LC0:
        .string "HELLORD"
        .text
        .globl  main
        .type   main, @function
main:
.LFB0:
        .cfi_startproc
        pushq   %rbp
        .cfi_def_cfa_offset 16
        .cfi_offset 6, -16
        movq    %rsp, %rbp
        .cfi_def_cfa_register 6
        movl    $.LC0, %edi
        call    puts
        movl    $0, %eax
        popq    %rbp
        .cfi_def_cfa 7, 8
        ret
        .cfi_endproc
.LFE0:
        .size   main, .-main
        .ident  "GCC: (GNU) 11.5.0 20240719 (Red Hat 11.5.0-11)"
        .section        .note.GNU-stack,"",@progbits
```

Machine Code Comparison

- X86_64

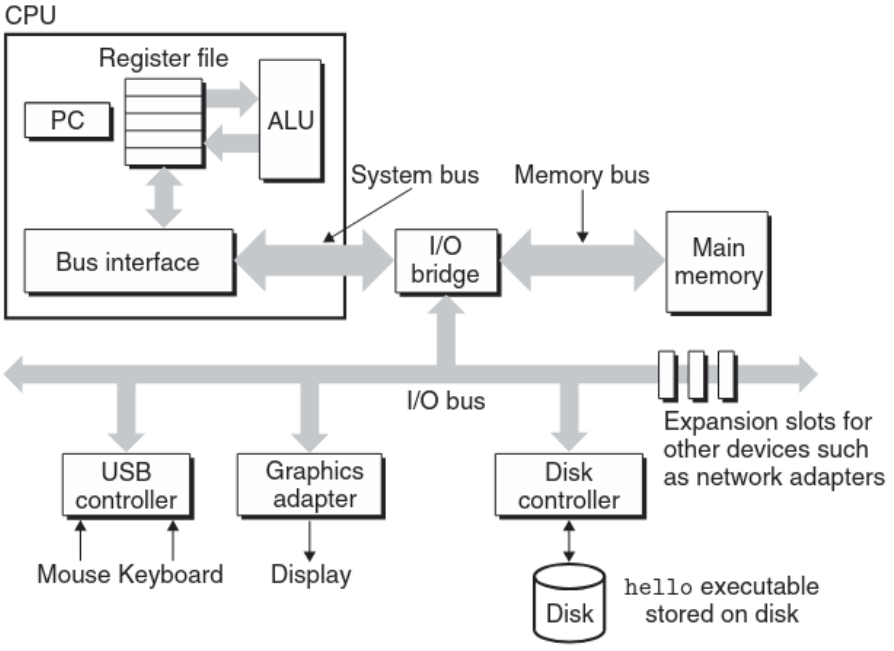
```
gcc -S hello.i -o hello.s
cat hello.s
        .file   "hello.c"
        .text
        .section        .rodata
.LC0:
        .string "HELLORD"
        .text
        .globl  main
        .type   main, @function
main:
.LFB0:
        .cfi_startproc
        pushq   %rbp
        .cfi_def_cfa_offset 16
        .cfi_offset 6, -16
        movq    %rsp, %rbp
        .cfi_def_cfa_register 6
        movl    $.LC0, %edi
        call    puts
        movl    $0, %eax
        popq    %rbp
        .cfi_def_cfa 7, 8
        ret
        .cfi_endproc
.LFE0:
        .size   main, .-main
        .ident  "GCC: (GNU) 11.5.0 20240719 (Red Hat 11.5.0-11)"
        .section        .note.GNU-stack,"",@progbits
```

RISC-V

```
cat hello.s
        .file   "hello.c"
        .option pic
        .text
        .section        .rodata
        .align  3
.LC0:
        .string "HELLORD"
        .text
        .align  1
        .globl  main
        .type   main, @function
main:
        addi    sp,sp,-16
        sd      ra,8(sp)
        sd      s0,0(sp)
        addi    s0,sp,16
        lla     a0,.LC0
        call    puts@plt
        li      a5,0
        mv      a0,a5
        ld      ra,8(sp)
        ld      s0,0(sp)
        addi    sp,sp,16
        jr      ra
        .size   main, .-main
        .ident  "GCC: (Ubuntu 11.4.0-1ubuntu1~22.04) 11.4.0"
        .section        .note.GNU-stack,"",@progbits
```

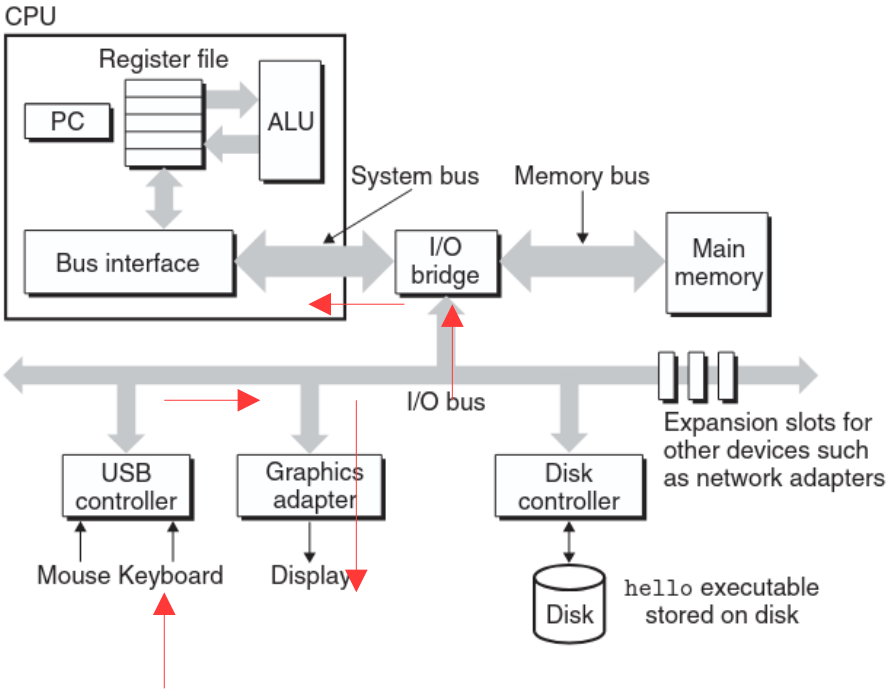
Modern Computer Architecture

- Layout focused on PC.
- Common layout.



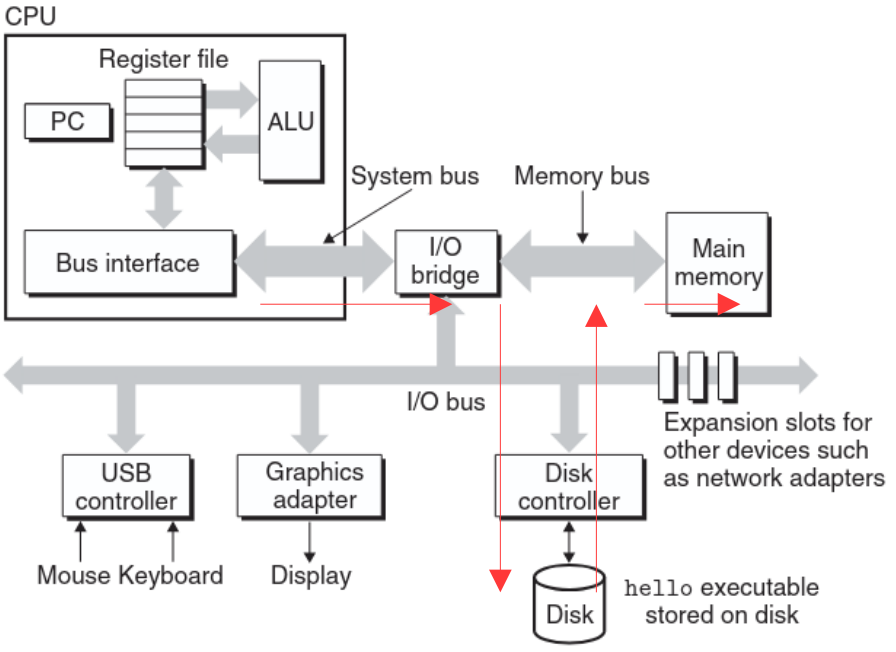
HELLO execution

- USB Controller.
  - ./hello from keyboard
  - ./hello echoed to display
- Command sent to “TTY”



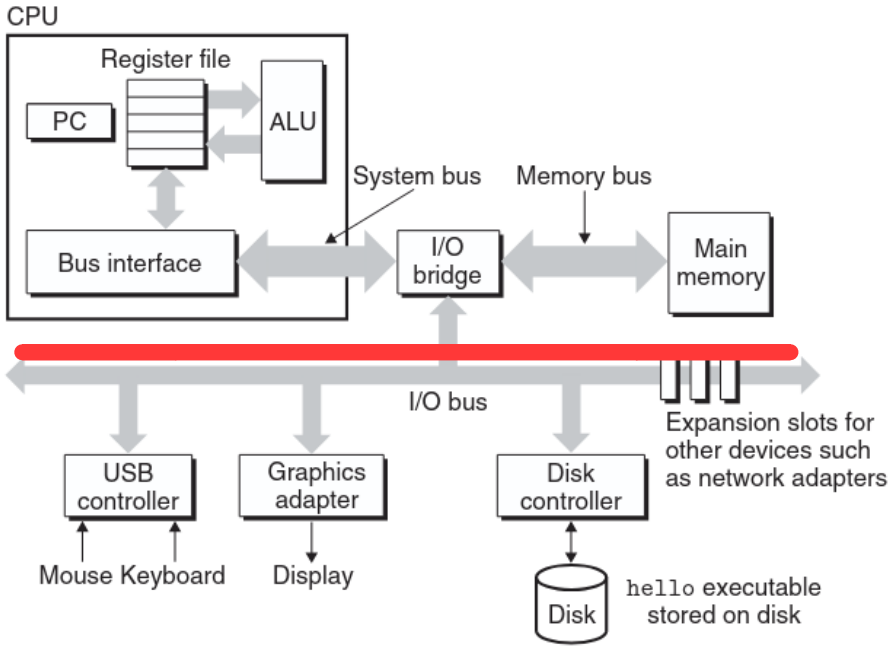
HELLO Execution Continued

- TTY is a program.
- TTY sends data to shell.
- Shell asks kernel to run.
- Kernel loads program.
- Program executes.



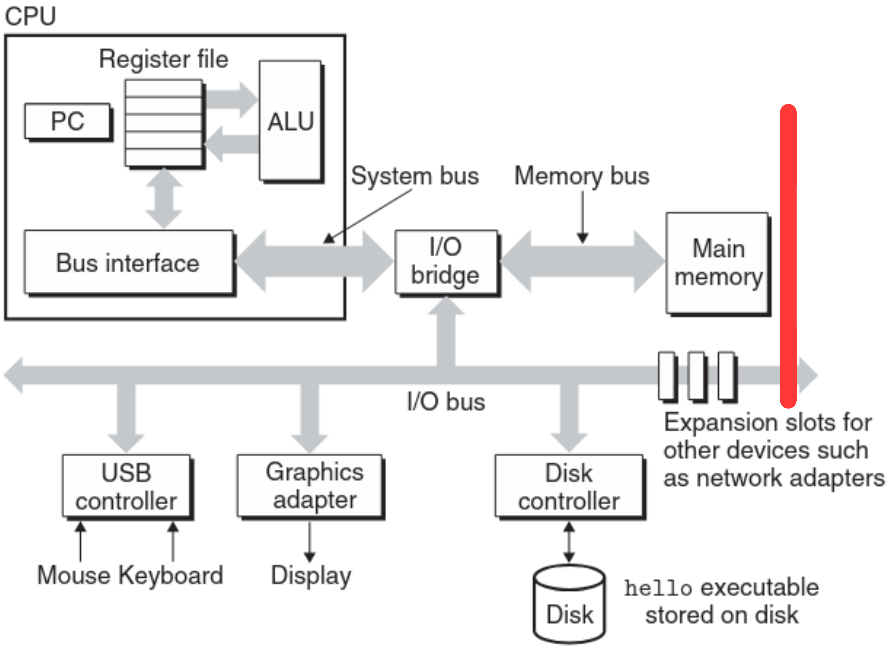
I/O Paths: Bus and Bridge

- Bus and Bridge
- PCI-E most common
- USB is bridged to PCI-E



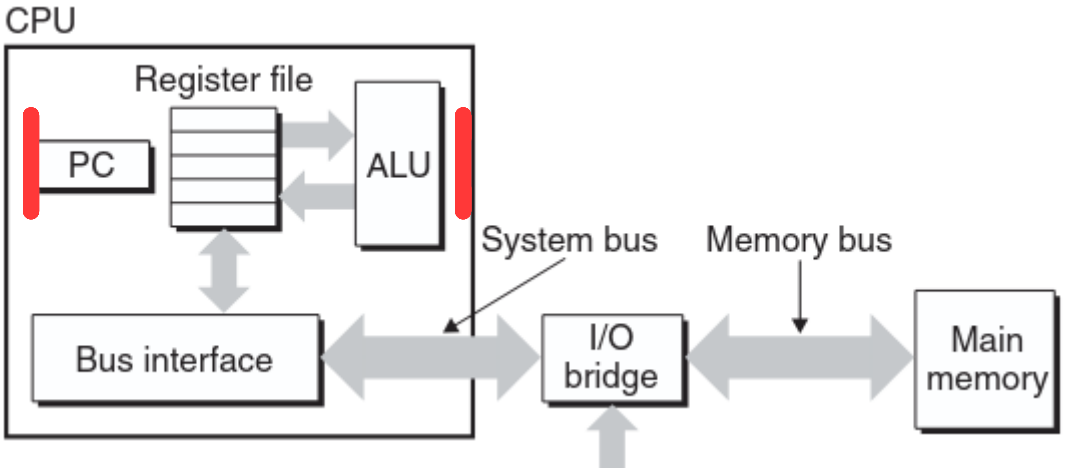
Main Memory

- Actual location of data
- Constrains performance
- Its why your PC is $$$



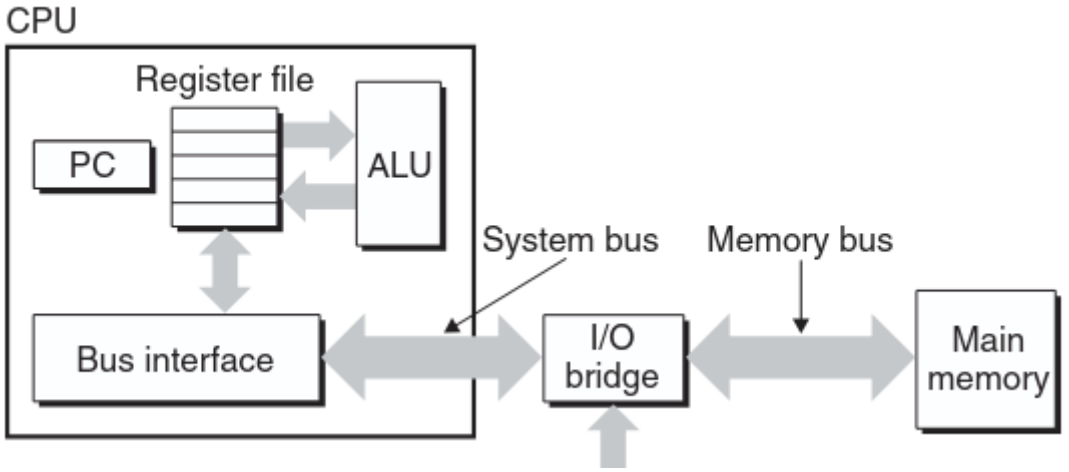
CPU: Structure

- PC: Program Counter
- ALU: Arithmetic Logic Unit



CPU: Basic Operations

- DMA: Direct Memory Access
- Load, Store, Operate, Jump



CPU: Layers of Caching

- Cost and Locality
- Caching is hard
- Page tables.

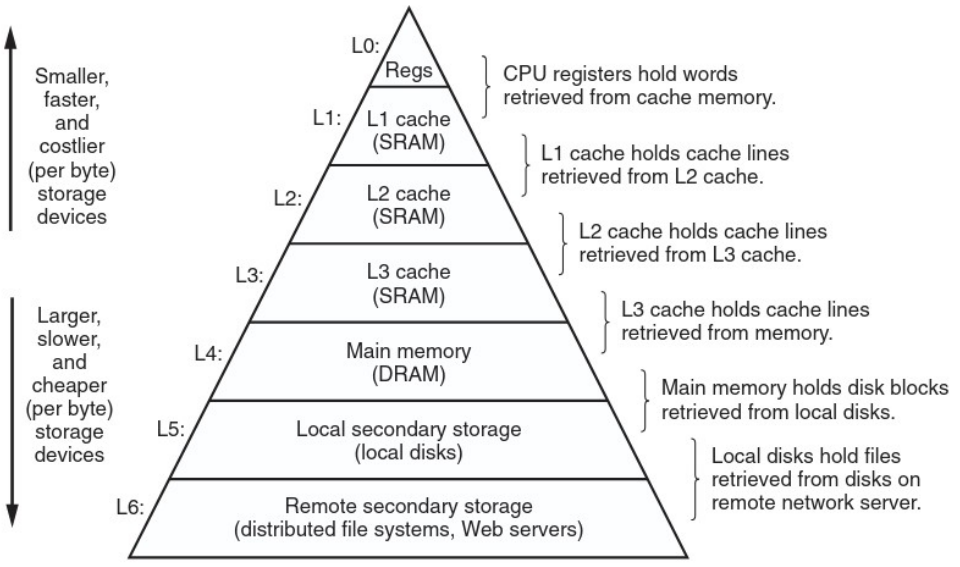
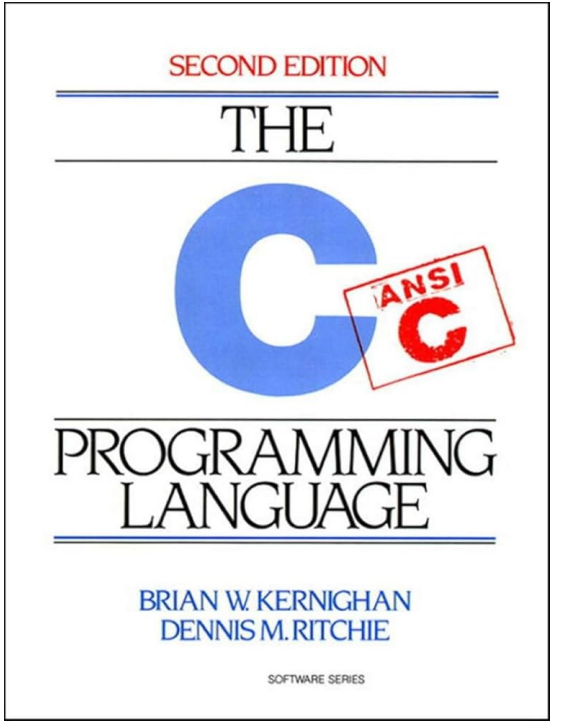


Figure 1.9 **An example of a memory hierarchy.**

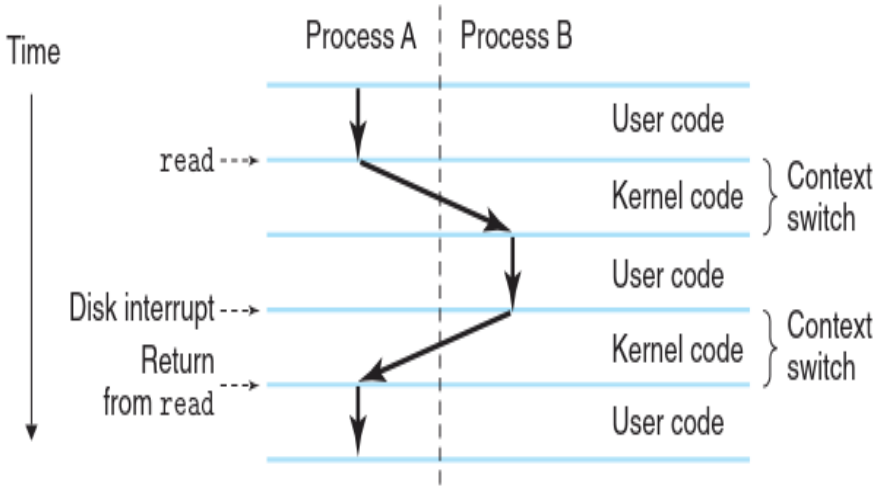
A nod to K&R

- Bell Labs
  - David Richie
  - Brian Kernighan
  - Ken Thompson
- Monopolies are bad(?)



The Kernel: Processes

- Time Slices
- Context Switches
- Process Schedulers

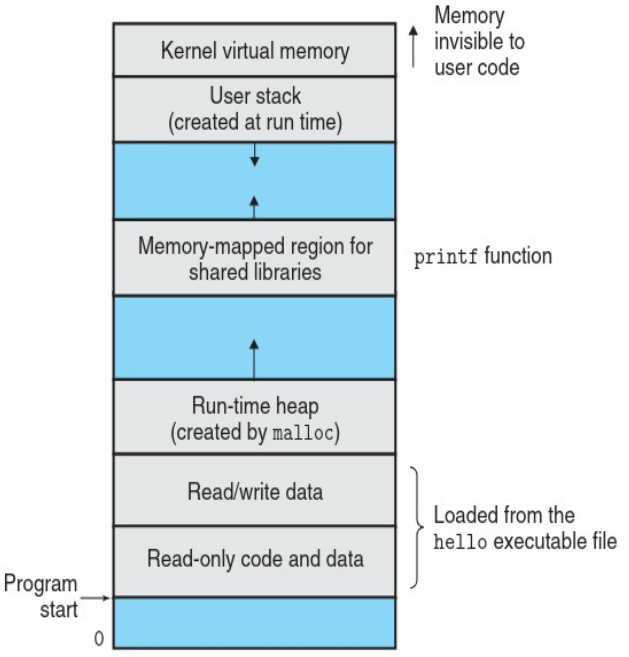


The Kernel: Threads

- Shared Structures
- Share Context
- Fun debugging
- https://en.wikipedia.org/wiki/Pthreads
  - Single file example of execution

The Kernel: Virtual Memory

- Process Stack
- Process Heap
- Mapping Tricks
  - Mapped to disk
  - Mapped to DRAM
  - Mapped to CACHE

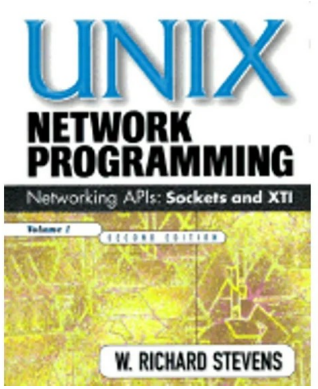


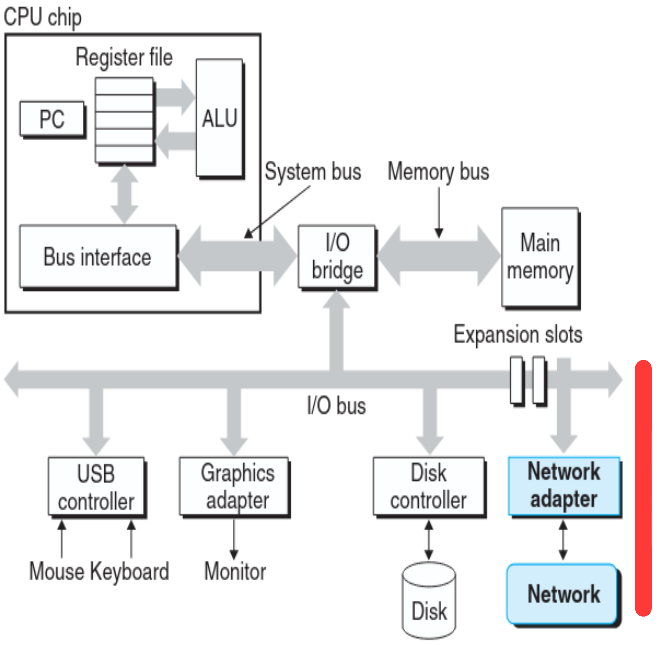
The Kernel: File Systems

- Types and targets
  - Linux: EXT4, ZFS, VFAT
  - Windows: NTFS
  - OS/X: APFS, HFS+
- https://en.wikipedia.org/wiki/File_system
- https://en.wikipedia.org/wiki/Computer_data_storage

The Kernel: Networking

- Just another I/O Device
- Network Stacks
  - TCP/IP
  - UDP
- W. Richard Stevens





Amdahl's Law: Performance

- Thought Experiment
  - alpha
  - K factor improvement
  - S Total Speed Up

$$S = \frac{1}{(1-\alpha) + \alpha/k}$$

- Improvement to a part of system can have negligible impact depending on alpha.