

CS 300 Leftovers

argc/argv

- “The C Programming Language” section 5.10, page 114

```
int main(int argc, char** argv)
```

argc - number of entries in argv

argv - array of character pointers containing the arguments

argv[0] - the name of the executable

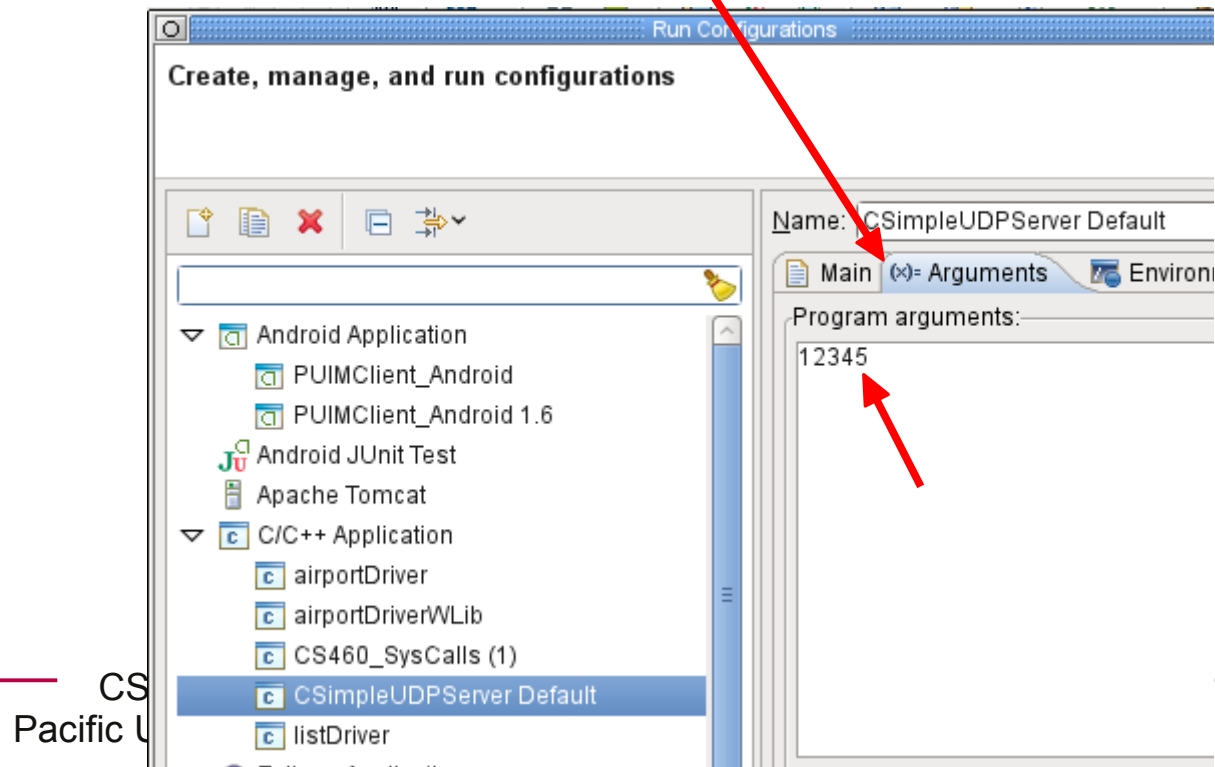
argv[1] - the first command line argument

./runMe argument

argc = 2

argv[0] - ./runMe

argv[1] - argument



man pages

- manual pages
 - man bash
 - man man
 - man ls

also available online
google → man bash
(may be different than what is on your machine)

```
MAN(1)                                Manual pager utils                                MAN(1)

NAME
    man - an interface to the on-line reference manuals

SYNOPSIS
    man [-c|-w|-tZ] [-H[browser]] [-T[device]] [-X[dpi]] [-adhu7V] [-i|-I]
    [-m system[...]] [-L locale] [-p string] [-C file] [-M path] [-P
    pager] [-r prompt] [-S list] [-e extension] [--warnings [warnings]]
    [[section] page ...] ...
    man -l [-7] [-tZ] [-H[browser]] [-T[device]] [-X[dpi]] [-p string] [-P
    pager] [-r prompt] [--warnings[warnings]] file ...
    man -k [apropos options] regexp ...
    man -f [whatis options] page ...

DESCRIPTION
    man is the system's manual pager. Each page argument given to man is
    normally the name of a program, utility or function. The manual page
    associated with each of these arguments is then found and displayed. A
    section, if provided, will direct man to look only in that section of
    the manual. The default action is to search in all of the available
    sections, following a pre-defined order and to show only the first page
    found, even if page exists in several sections.
```

<http://linux.die.net/man/>
<http://www.kernel.org/doc/man-pages/>

man pages - Library Function

FOPEN(3)

Name

fopen, fdopen, freopen - stream open functions

Synopsis

```
#include <stdio.h>
```

```
FILE *fopen(const char *path, const char *mode);
```

Description

(arguments or command line options are listed here)

Return Value

Errors

See Also

Referenced By

manual sections

- 0 Header files (usually found in /usr/include)
- 1 Executable programs or shell commands
- 2 System calls (functions provided by the kernel)
- 3 Library calls (functions within program libraries)
- 4 Special files (usually found in /dev)
- 5 File formats and conventions eg /etc/passwd
- 6 Games
- 7 Miscellaneous (including macro packages and conventions), e.g. man(7), groff(7)
- 8 System administration commands (usually only for root)
- 9 Kernel routines [Non standard]
p means POSIX!

```
chadd@bart:~> man nanosleep
Man: find all matching manual pages (set MAN_POSIXLY_CORRECT to avoid this)
* nanosleep (2)
  nanosleep (3p)
Man: What manual page do you want?
Man:
```

```
chadd@bart:~> export set MAN_POSIXLY_CORRECT=1
chadd@bart:~> man nanosleep
```

```
chadd@bart:~> man -S 3p nanosleep
```

OR
Add that line to the end of `~/.bashrc`

Kernighan's Law, p 85

- "Debugging is twice as hard as writing the code in the first place. Therefore, if you write the code as cleverly as possible, you are, by definition, not smart enough to debug it."

Chapter 2

Operating System Structures

OS Services

- User Interface
- Program Execution
- I/O Operation
- File System manipulation
- Communication
- Error detection

- Resource Allocation
- Accounting
- Protection/Security

User Interface to the OS

- Command Interpreter

- Command line
- Unix Shell
- C:\
- Mac Terminal

Often these are application programs and not part of the OS.

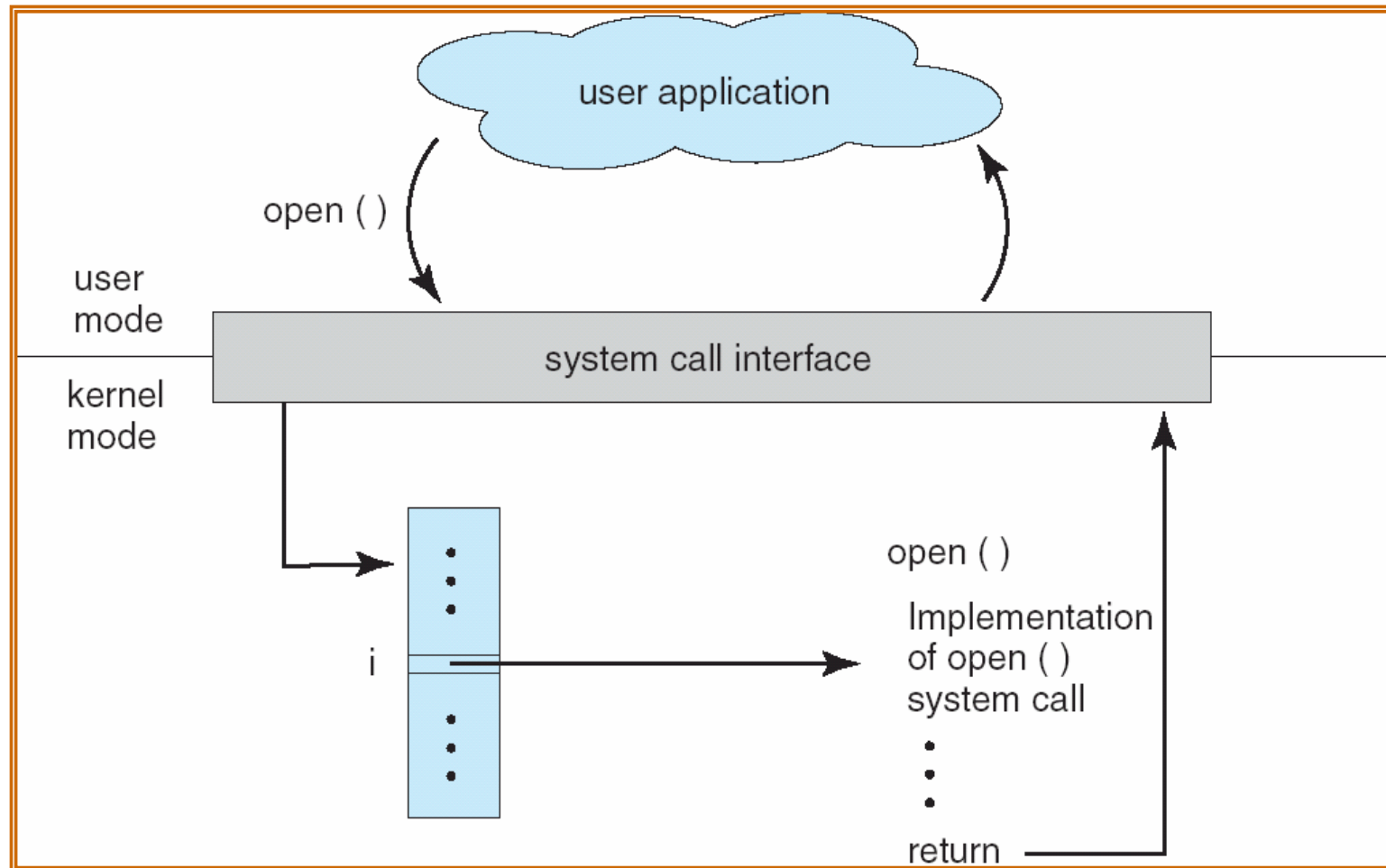
- GUI

- Xerox PARC
- Mac OS
- Windows
- X-Windows
- KDE/GNOME

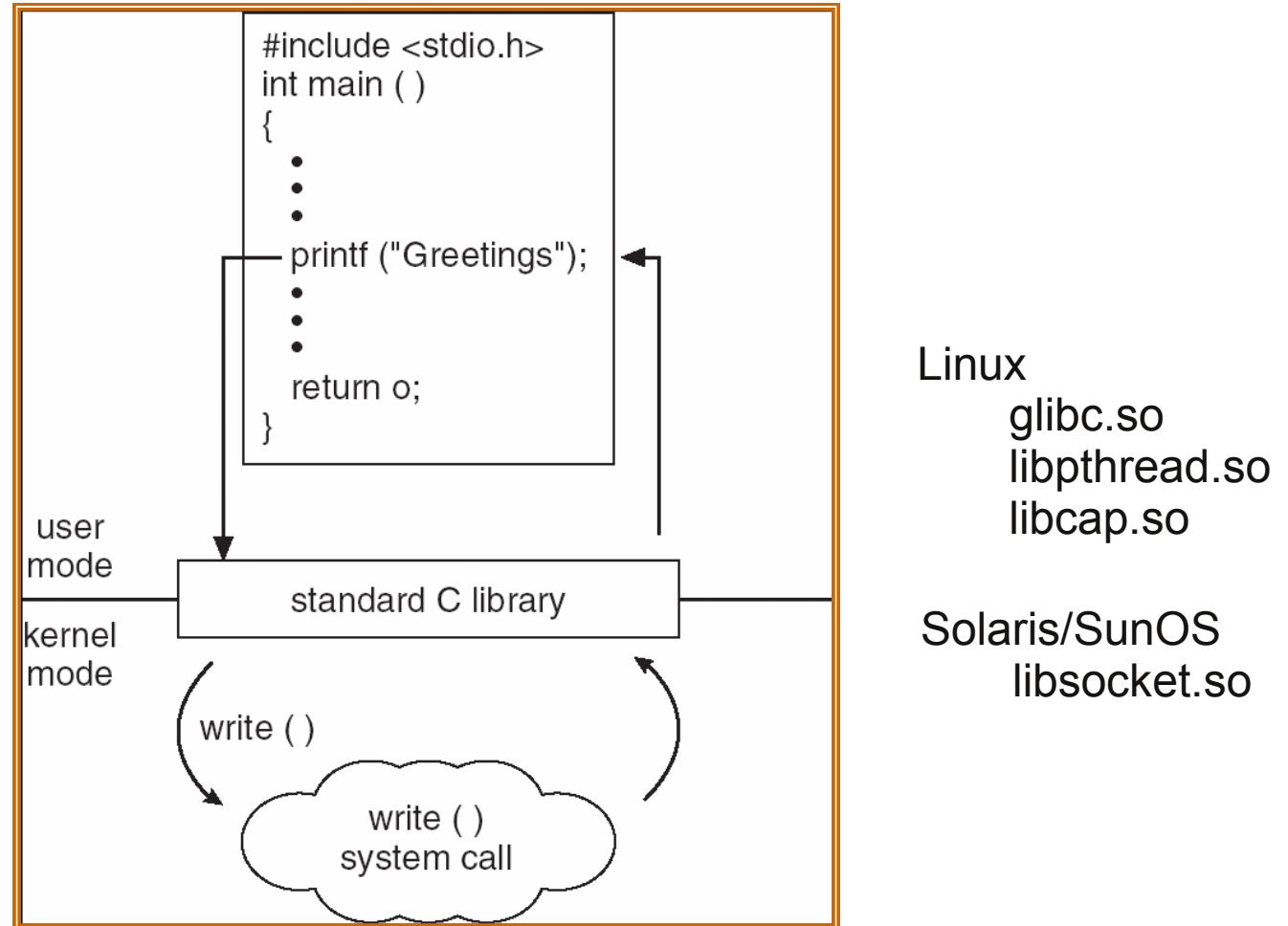
The true interface to the OS is via system calls

System Calls

- Interface to OS (kernel) services
- Wrapped in API (API = ?)
 - POSIX
libc.so
libgcc.so
 - Win32
 - Java API
 - why?



System Call via Library



Systems Calls: Data

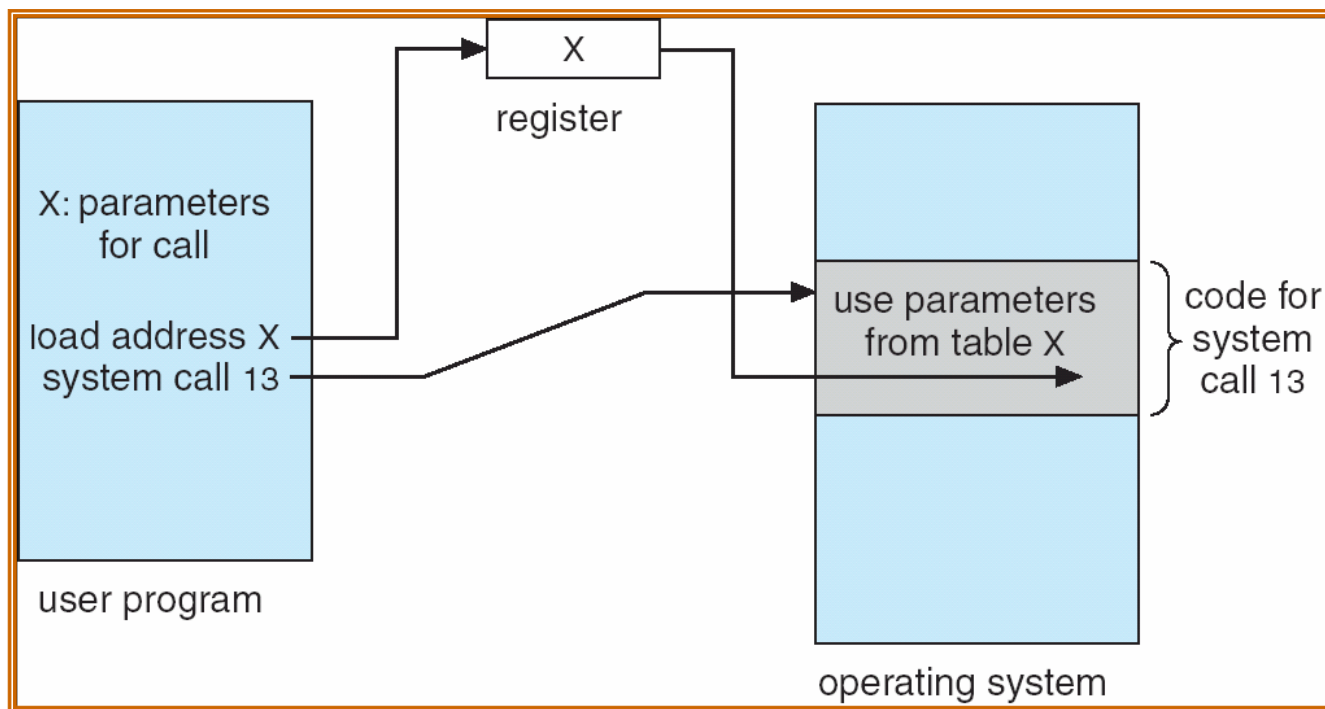
- Passing data to a system call

- Registers

- Block of memory

- Stack

- Advantages/
Disadvantages?



Types of System Calls

- Process Control
 - How does GDB work?
- File access
- Device access
- Information maintenance
- Communications

Process Control

- What are some process control system calls?
 - fork() / exec()

- How does GDB work?
 - the ptrace API

 - what does GDB need to do?

More System Calls....

- File Management

- Device Management
 - How is this different from File Management?
 - When would you use this?

Even More....

- Information Maintenance
 - Date
 - Time
- Communication
 - Message passing
 - pipes
 - Shared memory
 - Networking

“System Programs”

- System Utilities
 - common applications shipped with the OS/kernel
 - not necessarily part of the OS
 - often a wrapper around a system call
 - cp, mv, rm, cat
 - compilers

- Loaders

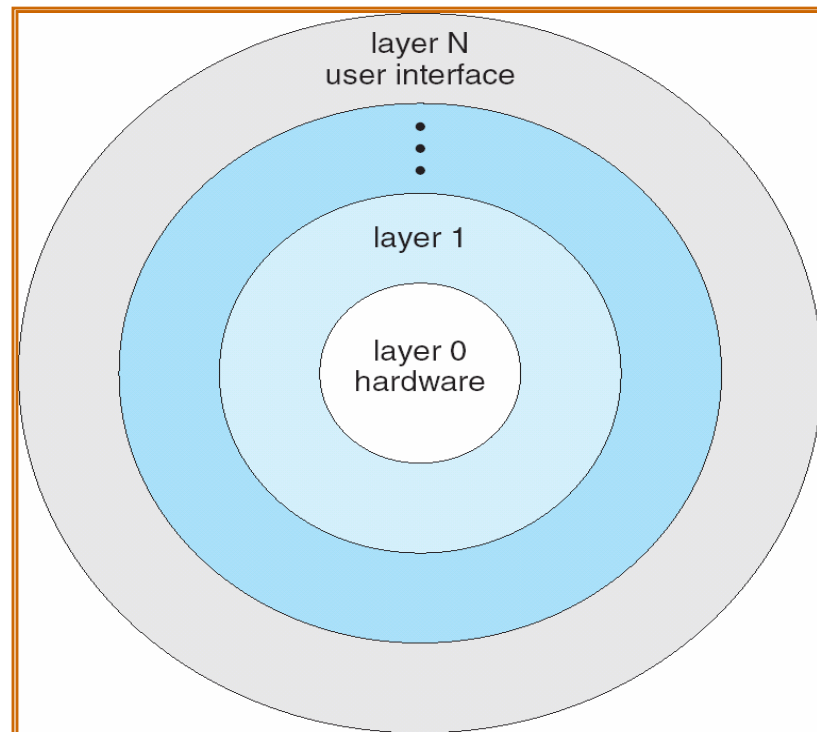
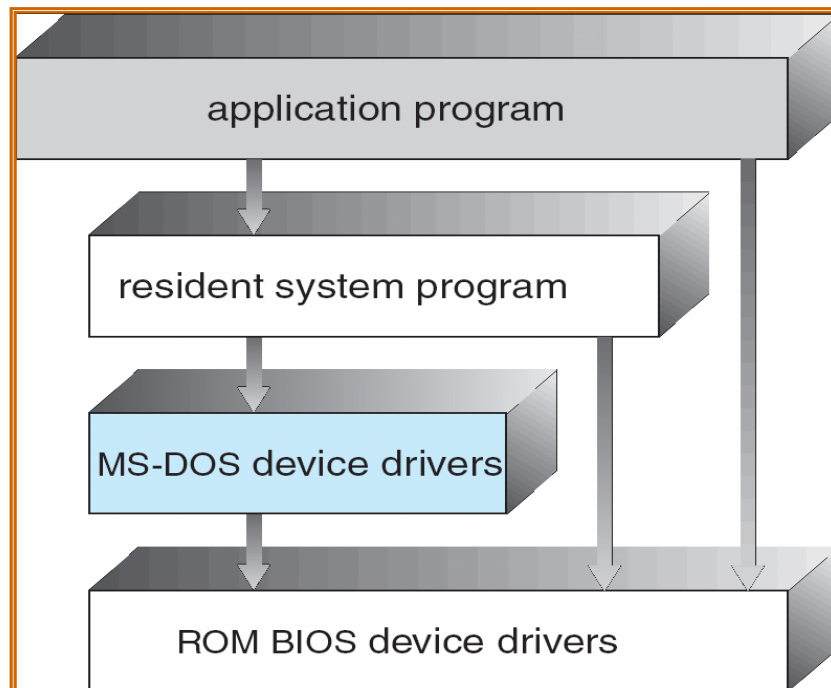
Operating System Design

- Design Goals
- Mechanism vs Policies
- Implementations
 - Assembly vs C
 - advantages/disadvantages?

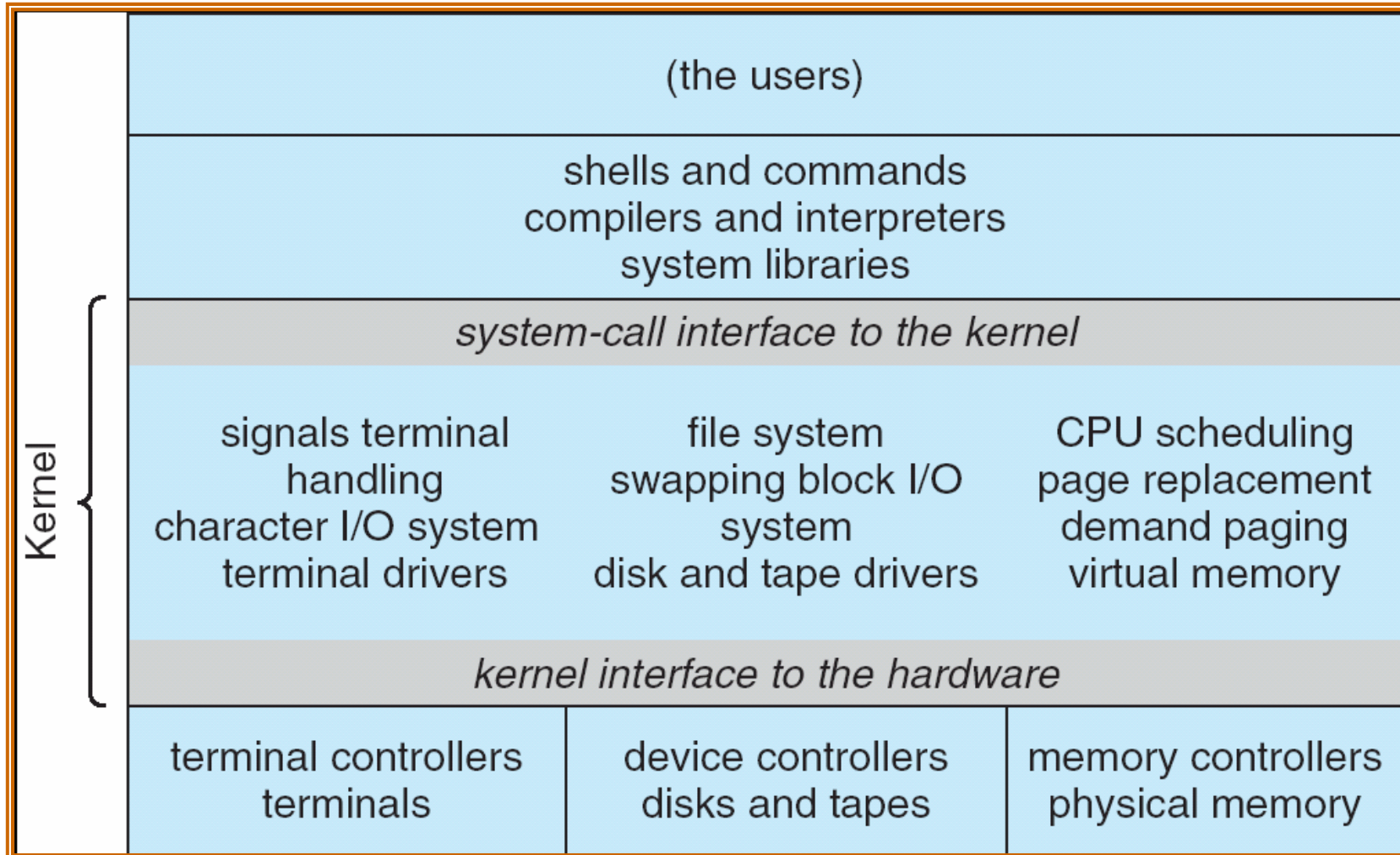
OS Structure

- Simple
 - MS DOS
 - Monolithic

- Layered



Old Unix



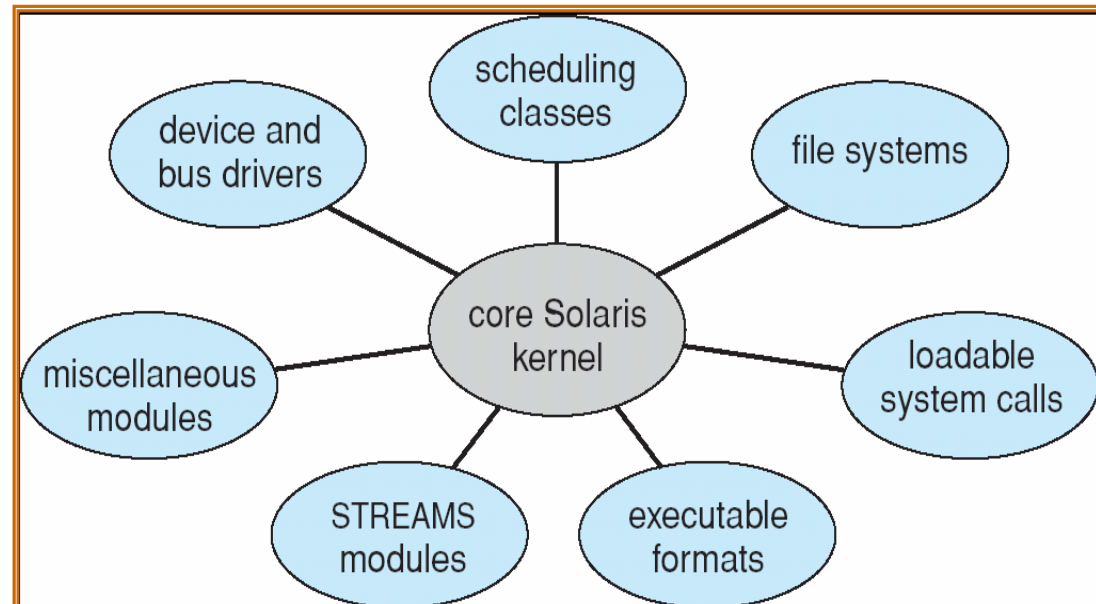
- Really Big Layers

Structure

- Microkernel
 - Mach/MacOS

- Modular

- Monolithic with Modules
 - Modern Linux



Virtual Machines (VM)

- Abstract away the hardware
 - Real or imagined hardware
 - Parallels
 - VMWare/Bochs
 - VirtualBox
 - Java VM
 - .Net

