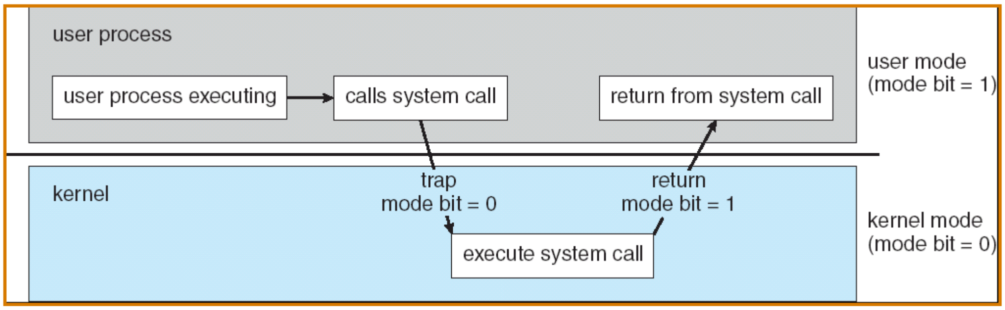
# Recap: Architecture Support in OS

| OS Service | Hardware Support |
|---|---|
| Protection | Kernel/user mode, protected instructions, base/limit registers |
| Interrupts | Interrupt vectors |
| System calls | Trap instructions and trap vectors |
| I/O | Interrupts |
| Scheduling, error recovery, accounting | Timer |
| Synchronization | Atomic instructions |
| Virtual memory | Translation look-aside buffers |

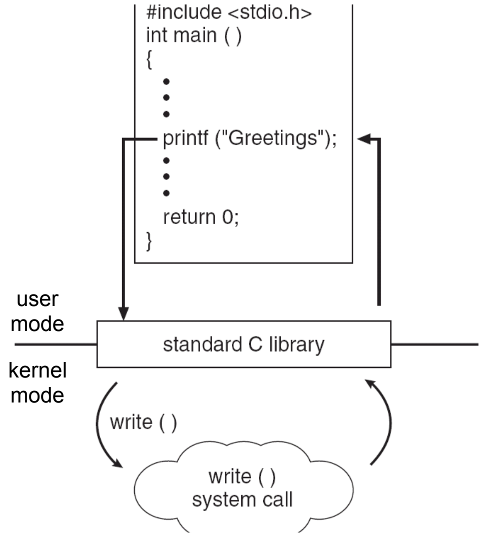
# Today: OS Services and Structure

- System call implementation
- Organization and components of an OS

# System Calls

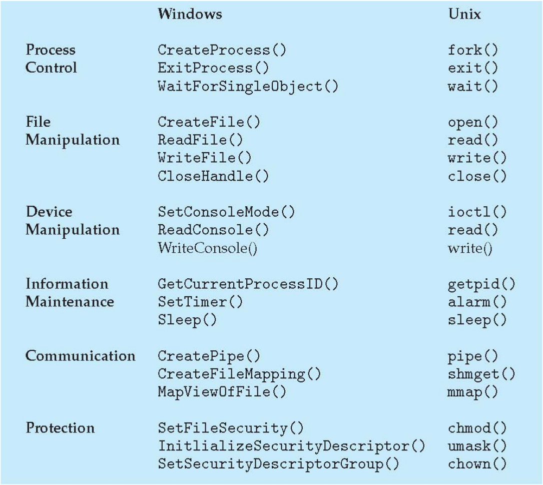


# Making a System Call



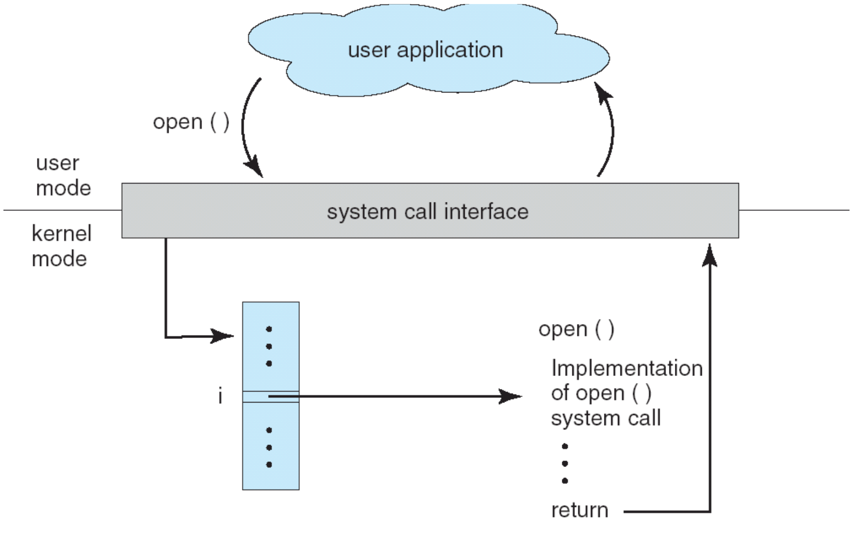
```
#include <stdio.h>
int main ( )
{
   •
   •
   •
   printf ("Greetings");
   •
   •
   •
   return 0;
}
```

# Example System Calls

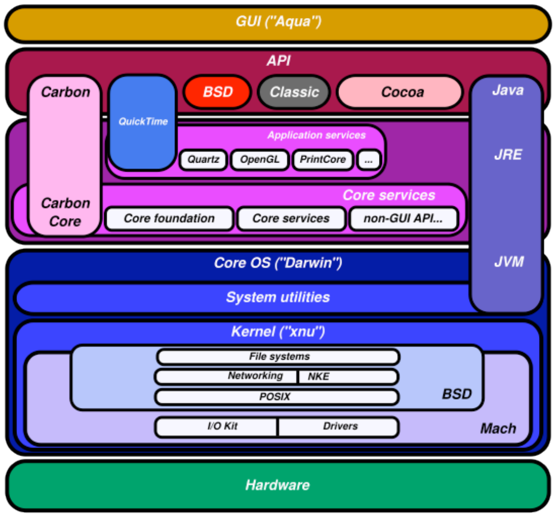


| | Windows | Unix |
|---|---|---|
| Process Control | CreateProcess()<br>ExitProcess()<br>WaitForSingleObject() | fork()<br>exit()<br>wait() |
| File Manipulation | CreateFile()<br>ReadFile()<br>WriteFile()<br>CloseHandle() | open()<br>read()<br>write()<br>close() |
| Device Manipulation | SetConsoleMode()<br>ReadConsole()<br>WriteConsole() | ioctl()<br>read()<br>write() |
| Information Maintenance | GetCurrentProcessID()<br>SetTimer()<br>Sleep() | getpid()<br>alarm()<br>sleep() |
| Communication | CreatePipe()<br>CreateFileMapping()<br>MapViewOfFile() | pipe()<br>shmget()<br>mmap() |
| Protection | SetFileSecurity()<br>InitlializeSecurityDescriptor()<br>SetSecurityDescriptorGroup() | chmod()<br>umask()<br>chown() |

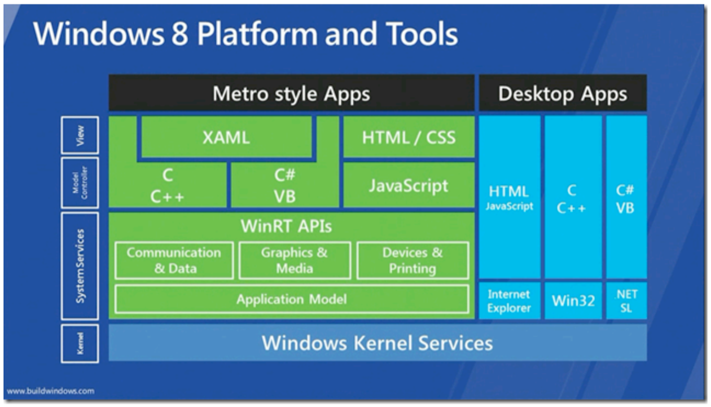
# System Call Implementation



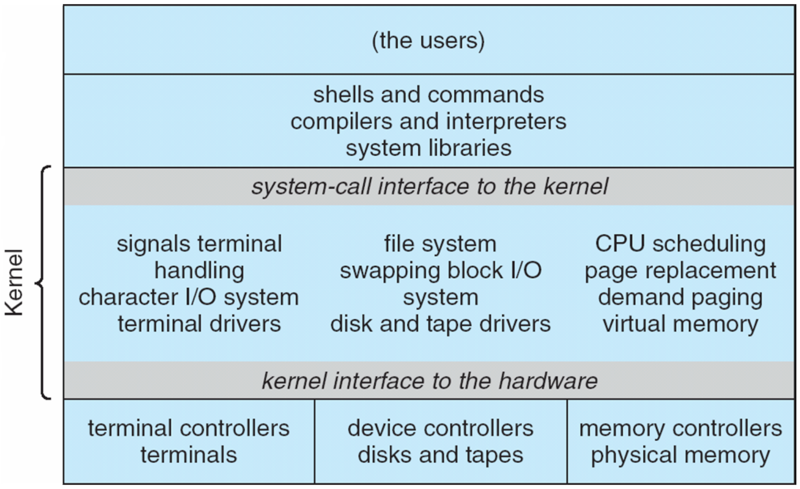
# OS Architecture: Mac OS X



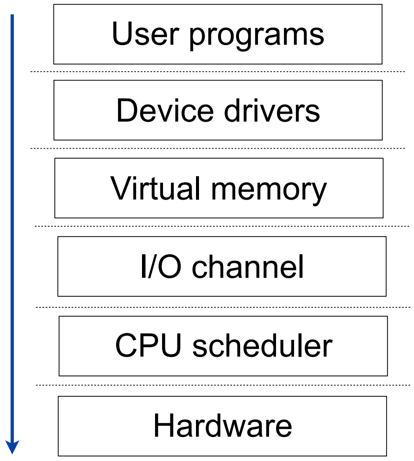
# OS Architecture: Windows 8



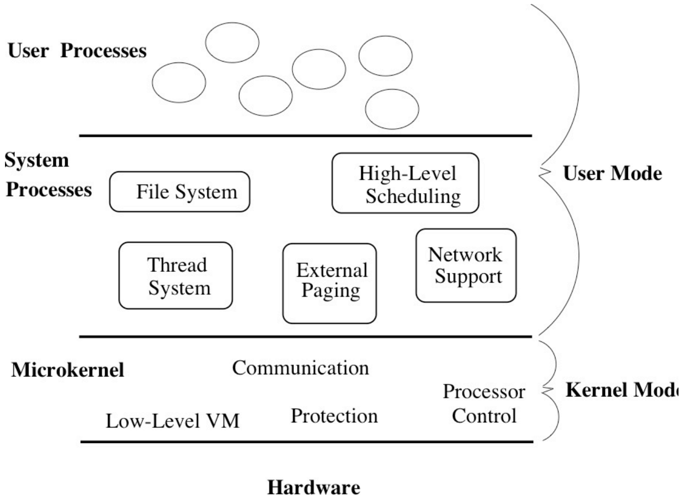
# Monolithic Kernel Design



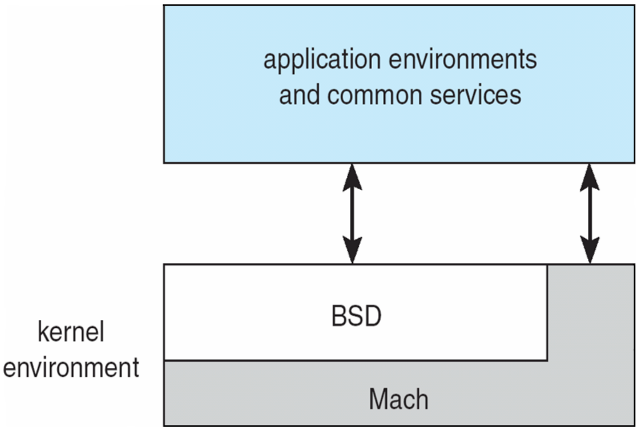
# Layered OS Design



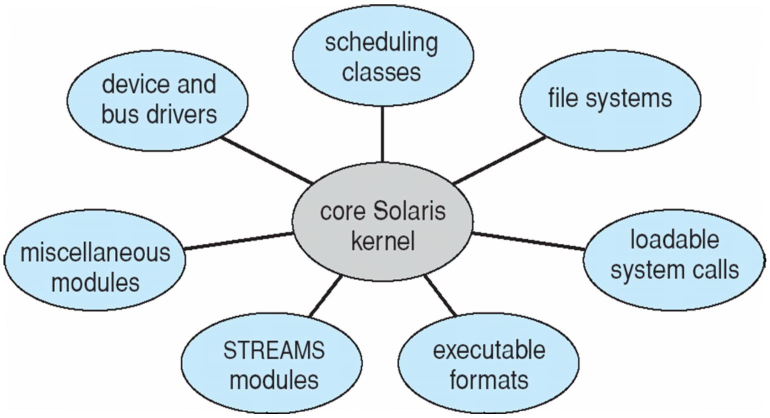
# Microkernel Design



# Mac OS X - hybrid approach



# Modules



# Summary

- System calls as interface to the OS
- OS design approaches
  - Monolithic
  - Layered
  - Microkernels
  - Modules
- Tradeoffs: simplicity vs. performance