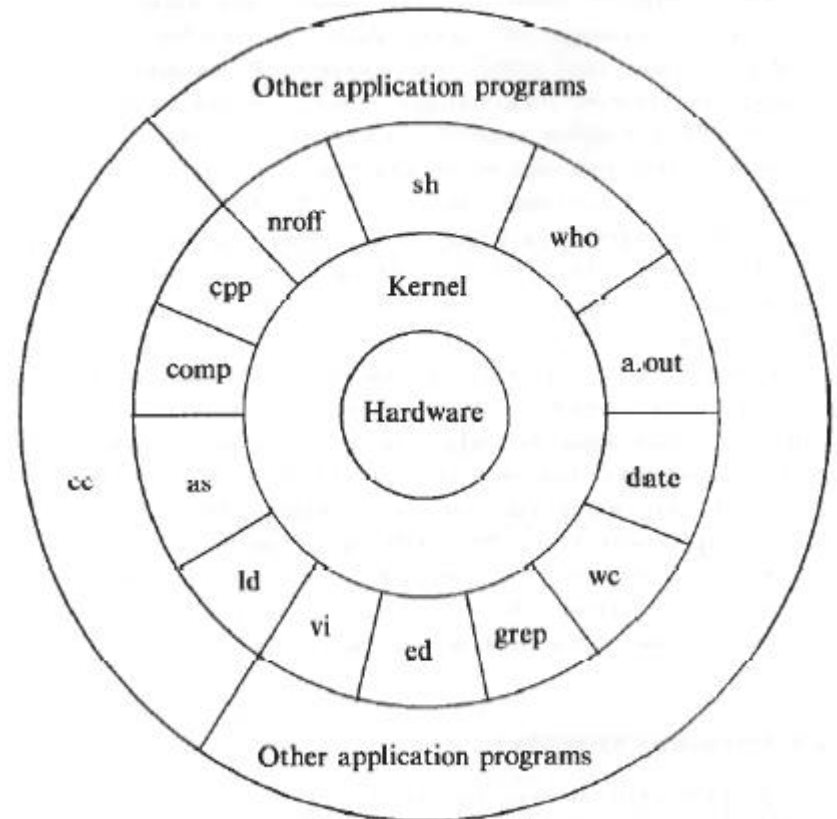


Unix Operating System Overview & Internals

Operating System

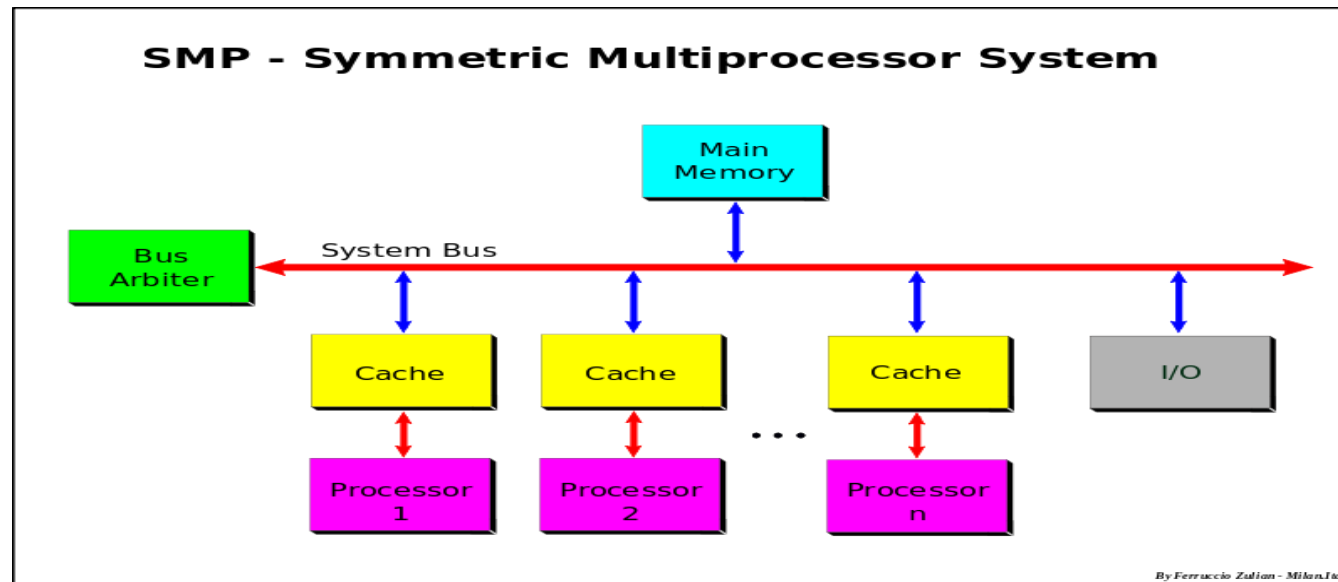
The high-level architecture of the UNIX System shown here.

- ❖ The hardware at the center of the diagram provides the operating system with basic services.
- ❖ The operating system interacts directly with the hardware, providing common services to programs
- ❖ The operating system is commonly called as system kernel or kernel , emphasizing its isolation from user programs
- ❖ The user programs are independent of the underlying hardware, its easy to move the user programs between different hardware if the programs doesn't assumptions about the underlying hardware



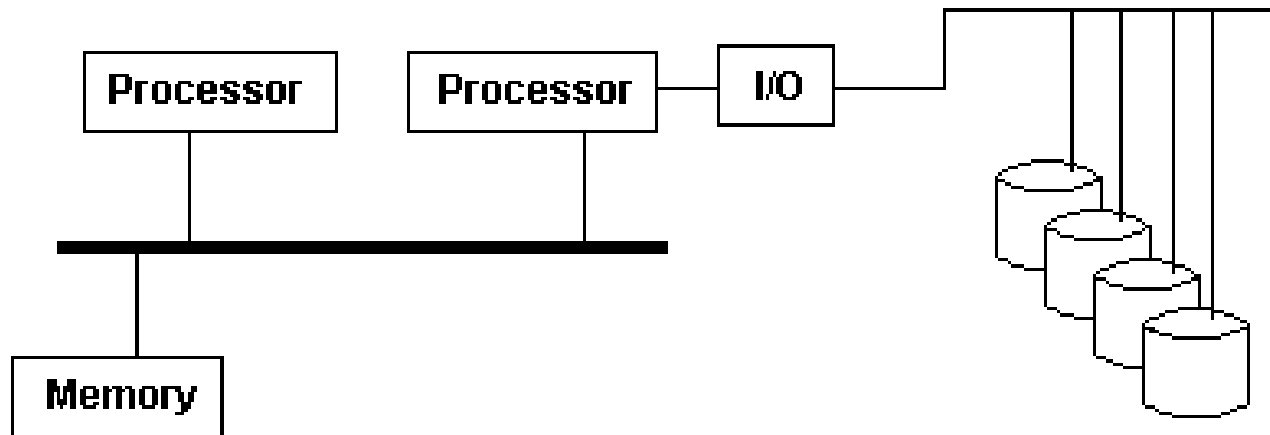
Multi Processing OS

- ❖ Multiprocessor Operating System refers to the use of two or more central processing units (CPU) within a single computer system. These multiple CPUs are in a close communication sharing the computer bus, memory and other peripheral devices. These systems are referred as tightly coupled systems.
- ❖ They are two different types, applied for various environments (a). Symmetric (b). Asymmetric
- ❖ Symmetric multiprocessing (SMP) involves a symmetric multiprocessor system hardware and software architecture where two or more identical processors connect to a single, shared main memory, have full access to all I/O devices, and are controlled by a single operating system instance that treats all processors equally, reserving none for special purposes. Most multiprocessor systems today use an SMP architecture.



Multi Processing OS Continued ..

- ❖ In an asymmetric multiprocessing system, not all CPUs are treated equally; for example, a system might only allow (either at the hardware or operating system level) one CPU to execute operating system code or might only allow one CPU to perform I/O operations
- ❖ Nowadays many operating systems supports SMP for scalable , better performance and throughput



Multi Core Processor

- ❖ A CPU, or Central Processing Unit, is what is typically referred to as a processor. A processor contains many discrete parts within it, such as one or more memory caches for instructions and data, instruction decoders, and various types of execution units for performing arithmetic or logical operations.
- ❖ A multiprocessor system contains more than one such CPU, allowing them to work in parallel. This is called SMP, or Simultaneous Multiprocessing.
- ❖ A multi*core* CPU has multiple execution cores on one CPU. Now, this can mean different things depending on the exact architecture, but it basically means that a certain subset of the CPU's components is duplicated, so that multiple "cores" can work in parallel on separate operations. This is called CMP, Chip-level Multiprocessing.
- ❖ For example, a multicore processor may have a separate L1 cache and execution unit for each core, while it has a shared L2 cache for the entire processor. That means that while the processor has one big pool of slower cache, it has separate fast memory and arithmetic/logic units for each of several cores. This would allow each core to perform operations at the same time as the others.
- ❖ There is an even further division, called SMT, Simultaneous Multithreading. This is where an even smaller subset of a processor's or core's component's is duplicated. For example, an SMT core might have duplicate thread scheduling resources, so that the core looks like two separate "processors" to the operating system, even though it only has one set of execution units. One common implementation of this is Intel's Hyper-threading.

Multi Core vs Multi Processor

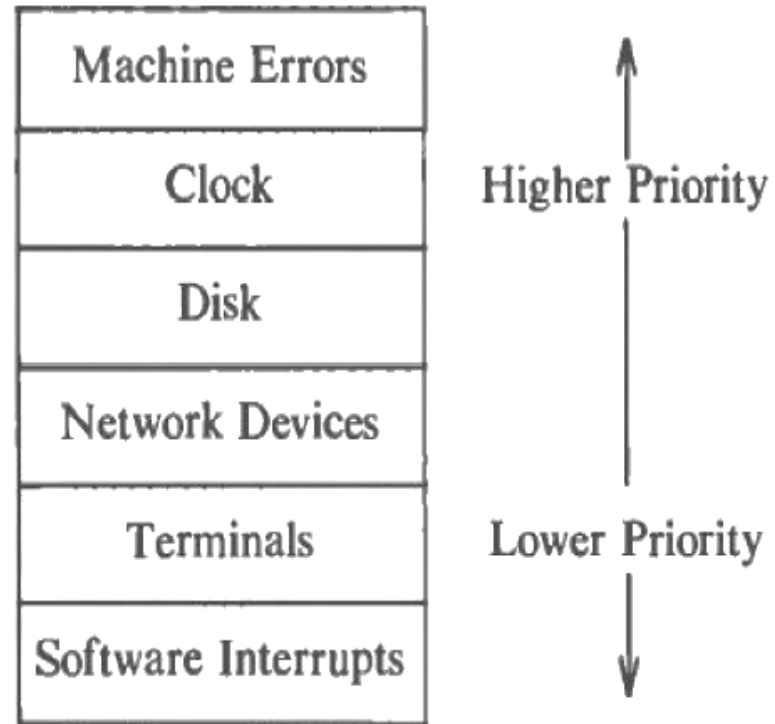
- ❖ Thus, you could have a multiprocessor, multicore, multithreaded system. Something like two quad-core (4 cpus), hyper-threaded processors would give you $2 \times 4 \times 2 = 16$ logical processors from the point of view of the operating system.
- ❖ Different workloads benefit from different setups. A single threaded workload being done on a mostly single-purpose machine benefits from a very fast, single-core/cpu system.
- ❖ In general, hardware these days is trending more and more toward highly parallel architectures, as most single CPU/core raw speeds are "fast enough" for common workloads across most models.
- ❖ CPUs have a clock speed – think of it as how fast the CPU does work. Means based each instruction will be executed based on the clock speed of the system
- ❖ For example, Intel's Core i5-3330 processor has a clock speed of 3 GHz and is a quad-core processor, which means it has four cores. All four cores in this Intel i5 processor are each running at 3 GHz.
- ❖ Another way 1 Core = 1 execution / 1 clock cycle
- ❖ (i.e) for 16 Core processor can execute 16 programs parallel

Multi Core vs Multi Processor Continued...

- ❖ Thus, you could have a multiprocessor, multicore, multithreaded system. Something like two quad-core (4 cpus), hyper-threaded processors would give you $2 \times 4 \times 2 = 16$ logical processors from the point of view of the operating system.
- ❖ Different workloads benefit from different setups. A single threaded workload being done on a mostly single-purpose machine benefits from a very fast, single-core/cpu system.
- ❖ In general, hardware these days is trending more and more toward highly parallel architectures, as most single CPU/core raw speeds are "fast enough" for common workloads across most models.
- ❖ CPUs have a clock speed – think of it as how fast the CPU does work. Means based each instruction will be executed based on the clock speed of the system
- ❖ For example, Intel's Core i5-3330 processor has a clock speed of 3 GHz and is a quad-core processor, which means it has four cores. All four cores in this Intel i5 processor are each running at 3 GHz.
- ❖ Another way 1 Core = 1 execution / 1 clock cycle
- ❖ (i.e) for 16 Core processor can execute 16 programs parallel

Operating System Services

- ❖ Process Management; Controlling the execution of process by allowing their creation, termination or suspension and communication
- ❖ Scheduler; Scheduling process fairly for execution on the CPU. Processes share the CPU in time-shared manner; CPU executes a process, kernel suspends it when time quantum elapses, and kernel schedules another process to execute. The kernel reschedules the suspended process
- ❖ Memory Management
- ❖ IO Management
- ❖ Different interrupt and its priorities



Sample Process Output

```
[sankar@localhost Cprogs]$ vi process.c
#include <stdio.h>
int main()
{
    while(1) {
        printf("Jataayu Research & Development Centre Welcomes You..!!\n");
    }
    return 0;
}
```

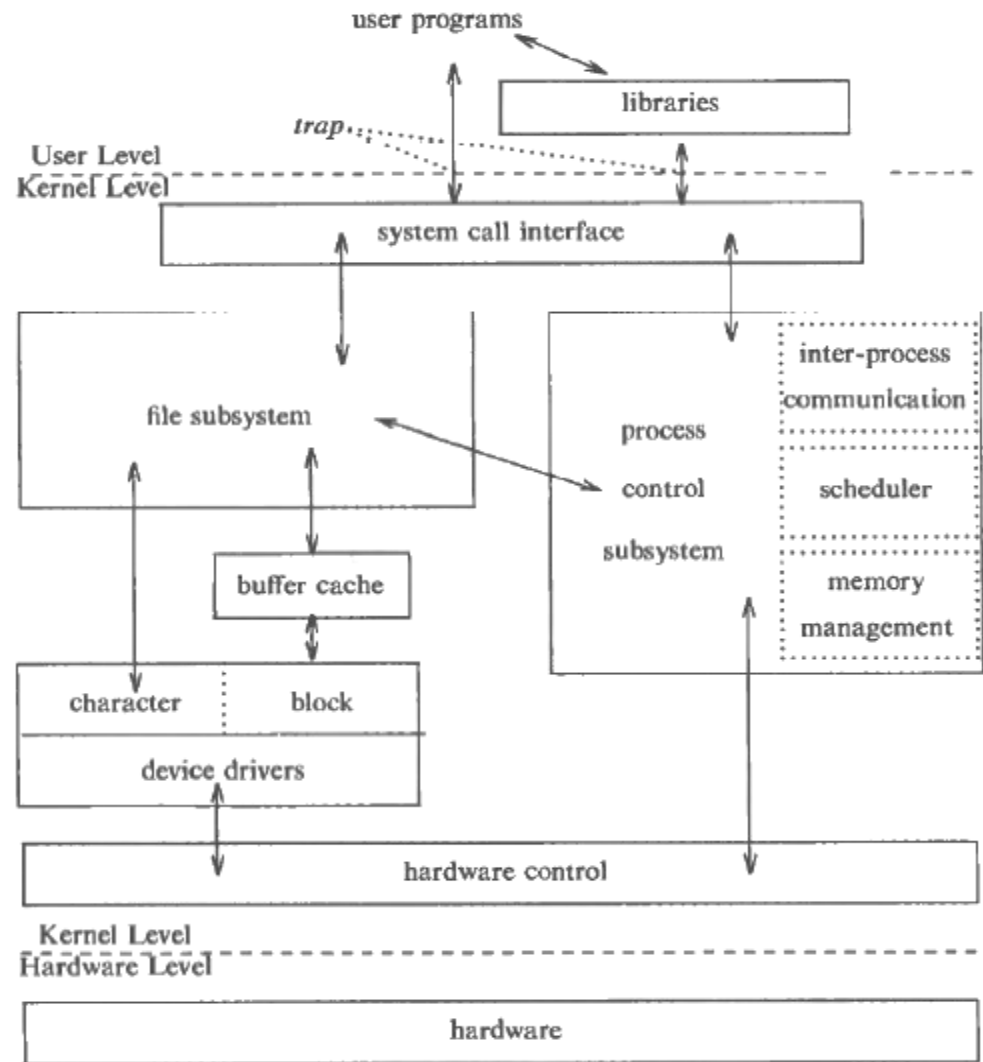
```
[sankar@localhost Cprogs]$ gcc process.c
[sankar@localhost Cprogs]$ ./a.out
Jataayu Research & Development Centre Welcomes You..!!
Jataayu Research & Development Centre Welcomes You..!!
Jataayu Research & Development Centre Welcomes You..!!
```

Login to root user :

```
[root@localhost ~]# cd /proc
[root@localhost proc]# ps -aef | grep a.out
sankar  2040  1870  1 11:40 pts/0  00:00:00 ./a.out
root    2043  2000  0 11:40 pts/1  00:00:00 grep a.out
[root@localhost proc]# ls 2040/
attr  clear_refs  cpuset  exe  io  maps  mounts  oom_adj  personality  schedstat  stack  status  wchan
auxv  cmdline    cwd    fd  limits  mem  mountstats  oom_score  root  sessionid  stat  syscall
cgroup  coredump_filter  environ  fdinfo  loginuid  mountinfo  net  pagemap  sched  smaps  statm  task
[root@localhost proc]#
```

Architecture of Unix OS

- ❖ High level block diagram of kernel shown here
- ❖ Represented by three levels
 - 1) User
 - 2) Kernel
 - 3) Hardware
- ❖ Sub-systems
 - 1) Process Subsystem
 - 2) File subsystem



File Subsystem

- ❖ High level block diagram of kernel shown here
- ❖ Represented by three levels
 - 1) User
 - 2) Kernel
 - 3) Hardware
- ❖ Sub-systems
 - 1) Process Subsystem
 - 2) File subsystem
 - 3) Memory Management
 - 4) Scheduler
 - 5) IO Module (typically hardware)

