

OPERATING SYSTEM

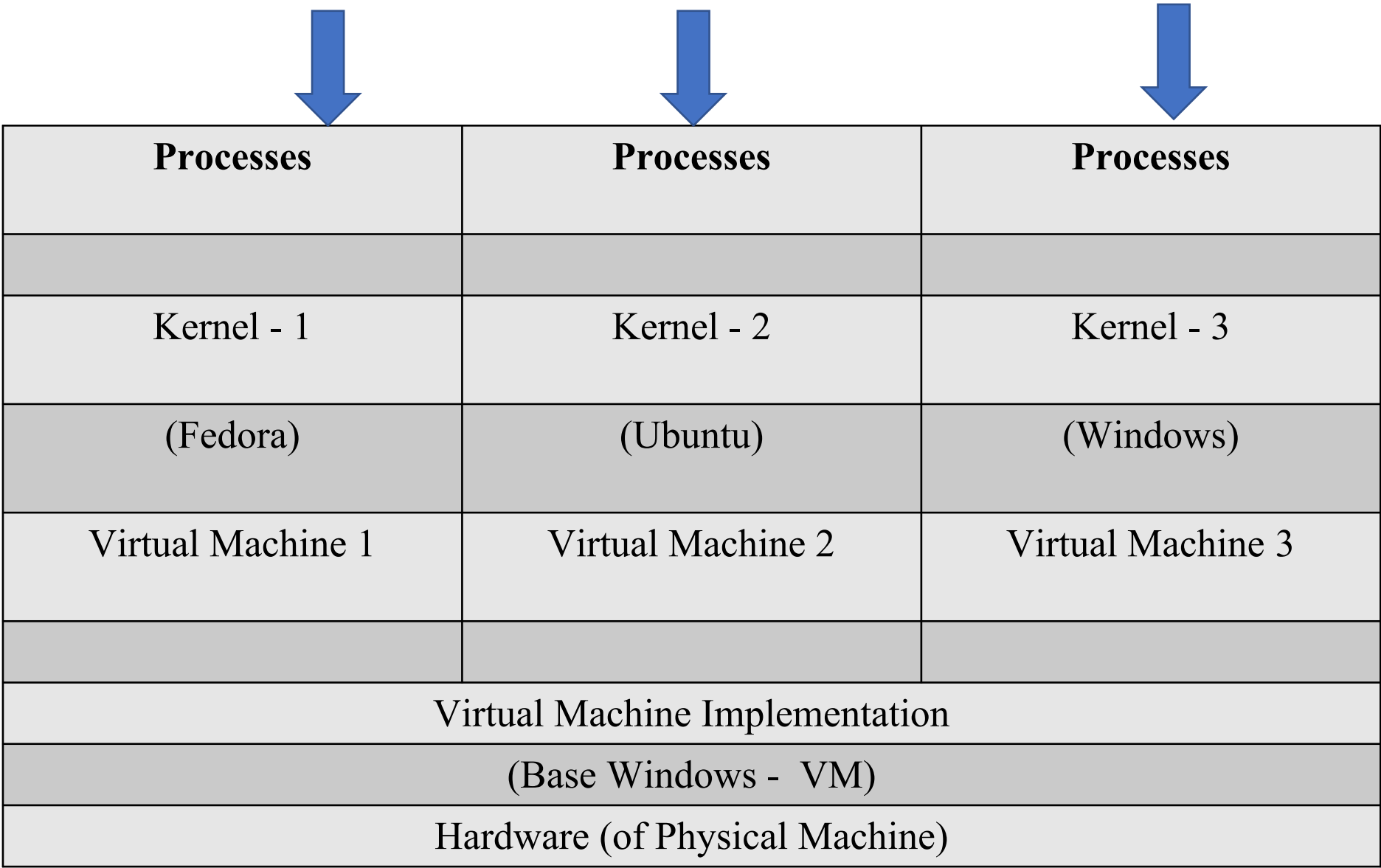
By :- Shripad Bhide

Execution Environment :- No. 1

No. 2

No. 3

Virtual Machine



When Hardware (CPU, Memory, NIC , Disk Drive etc...) of a single machine / computer used in several different execution environment , creating illusion (seems) that each separate execution environment is running its own private execution.

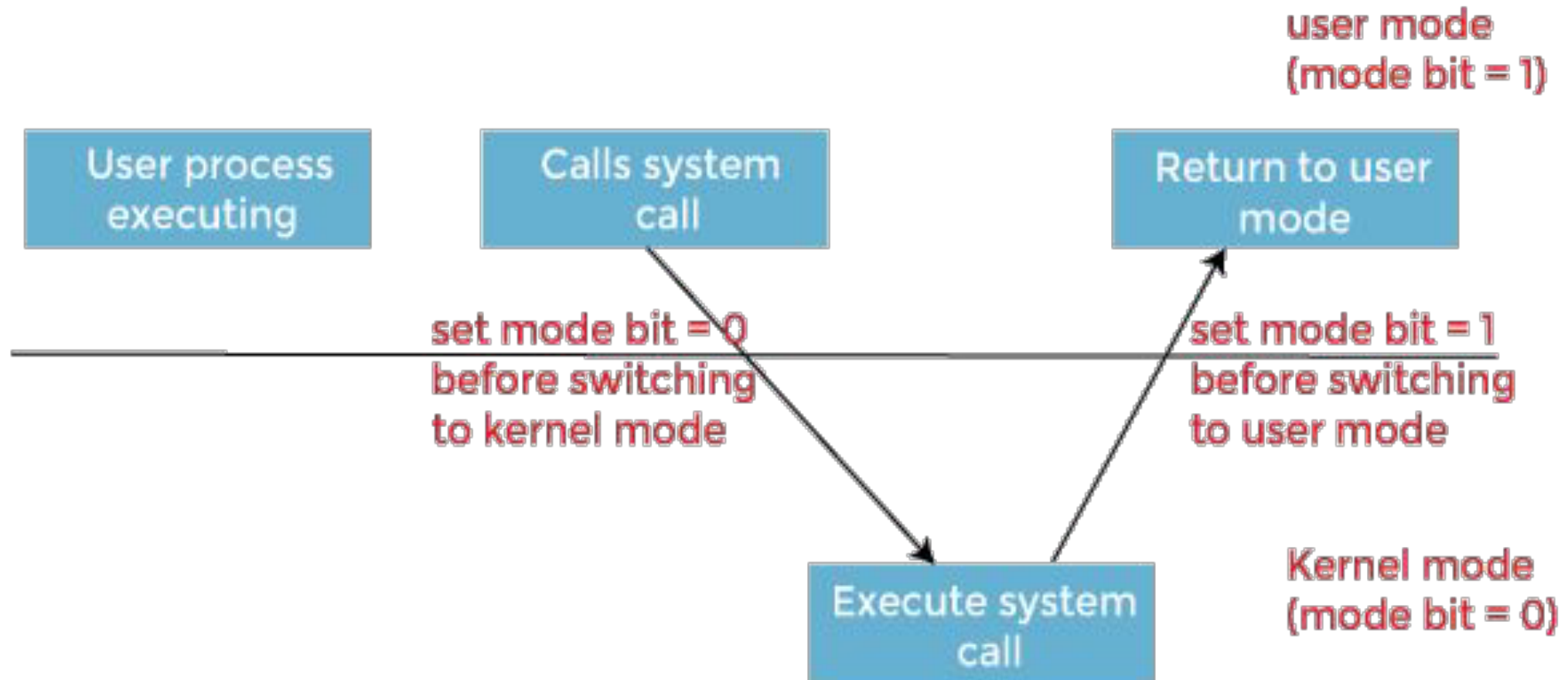
User Mode

When we are working with user applications like word, excel, power-point or any application program, then the system is in user mode.

Kernel Mode

When the user application is in demand for a service from the operating system or an interrupt occurs or [system call](#), then there will be a transition from user to kernel mode to fulfill the requests. When the system boots, the hardware starts in kernel mode and when the operating system is loaded, it starts user application in user mode.

To provide protection to the hardware, we have privileged instructions which execute only in kernel mode.



Virtual Machine Software runs in Kernal Mode. But Virtual Machine Itself runs in User Mode. (of Physical system)

Each Virtual Machine having its own Virtual User Mode and Virtual Kernel Mode and both runs in Physical User Mode.

Isolation of each virtual machine w.r.t basic hardware is maintained. Part of a disk is created is called Mini Disk. So parallel working with disk is possible.

e.g. Virtual Box , Vmware etc.

Virtual User
Mode1

Virtual Kernel
Mode1

Virtual Machine-1

Virtual User
Mode2

Virtual Kernel
Mode2

Virtual Machine-2

Virtual User
Mode3

Virtual Kernel
Mode3

Virtual Machine-3

User Mode of Physical Operating System

Kernel Mode of Physical Operating System

Virtual Machine
Software

Kernel :- Central component of an OS, Heart of an OS , Interface between user and software system.

It is an interface between Hardware components and software applications.

Applications like games , video, photos or audio editing , Network analysis etc. We install different softwares. e.g. Audio file require software for altering speakers , Gams required joysticks to work , for Photo editing photo-shop i.e. software demands hardware to function.

Kernel allows data to flow to the memory. It also decides order of program execution.

Separate space in memory is for kernel functions.

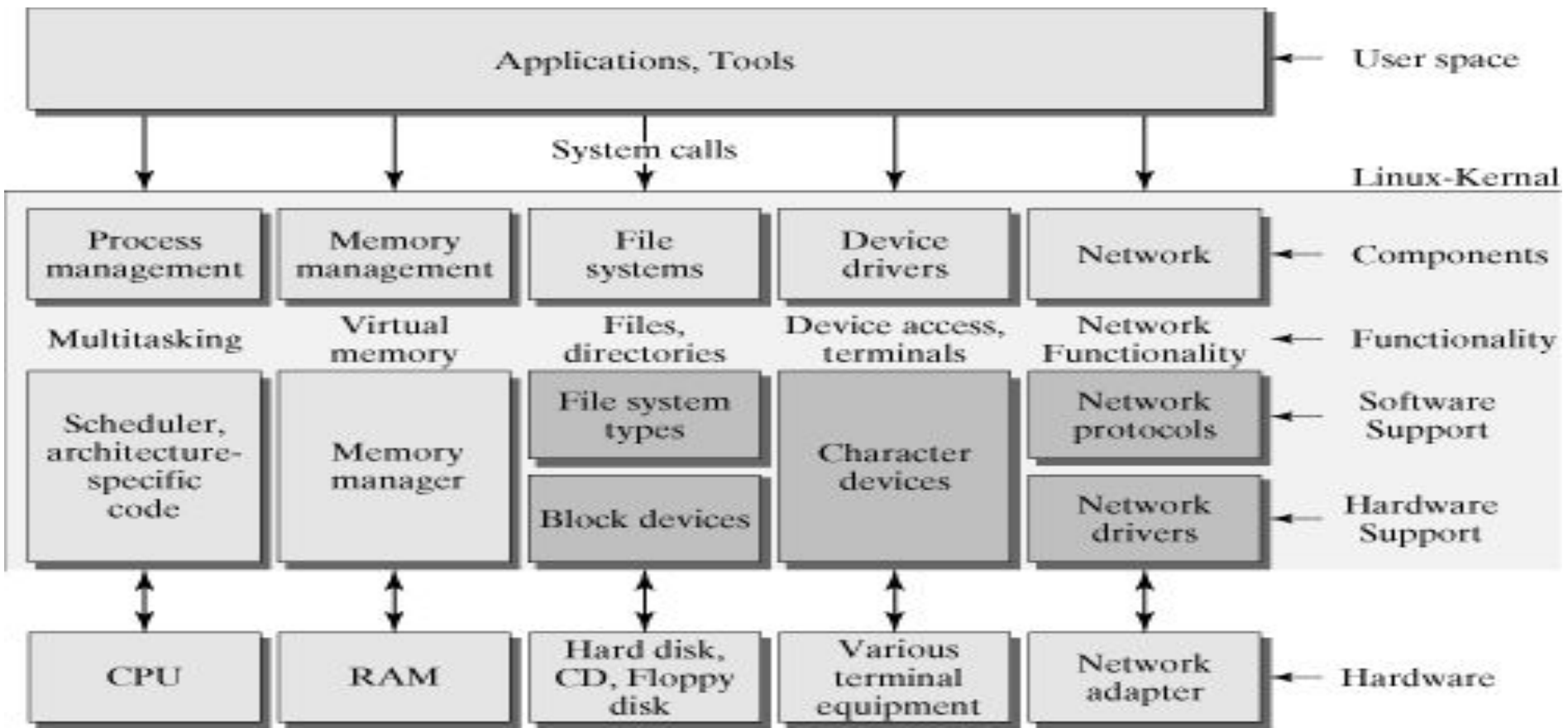
If user data **interferes** with kernel data System fails.

Process demands resources from kernel is system call.

A kernel is a central component of an operating system. It acts as an interface between the user applications and the hardware. The sole aim of the kernel is to manage the communication between the software (user level applications) and the hardware (CPU, disk memory etc). The main tasks of the kernel are :

Types Of Kernels

Kernels may be classified mainly in two categories :- i) Monolithic ii) Micro Kernel



There is a difference between kernel and OS. Kernel as described above is the heart of OS which manages the core features of an OS while if some useful applications and utilities are added over the kernel, then the complete package becomes an OS. So, it can easily be said that an operating system consists of a kernel space and a user space.

We can say that Linux is a kernel as it does not include applications like file system utilities, windowing systems and graphical desktops, system administrator commands, text editors, compilers etc. So, various companies add these kind of applications over linux kernel and provide their operating system like ubuntu, suse, CentOS, redHat etc.

Monolithic Kernel

In monolithic Kernels both user services and the kernel services are implemented in the same memory space. By doing this, the size of the Kernel as well as the size of the Operating System is increased. As there is no separate User Space and Kernel Space, so the execution of the process will be faster in Monolithic Kernels.

Advantages :-

- It provides CPU scheduling, memory scheduling, file management through System calls.
- Execution of the process is fast as there is no separate space.

Disadvantages :-

- If the service fails, then the system failure happens.
- If you try to add new services then the entire Operating System needs to be modified.

Microkernel :-

A Microkernel is not the same as the Monolithic kernel. It is a little bit different because in a Microkernel, the user services and kernel services are implemented into different spaces. Because of using User Space and Kernel Space separately, it reduces the size of the Kernel and in turn, reduces the size of the Operating System.

As we are using different spaces for user and kernel service, the communication between application and services is done with the help of message parsing because of this it reduces the speed of execution.

The **advantage** of microkernel is that it can easily add new services at any time.

The **disadvantage** of microkernel is that here we are using User Space and Kernel Space separately. So, the communication between these can reduce the overall execution time.

Remaining types - Hybrid Kernel , Nanokernel , Exokernel

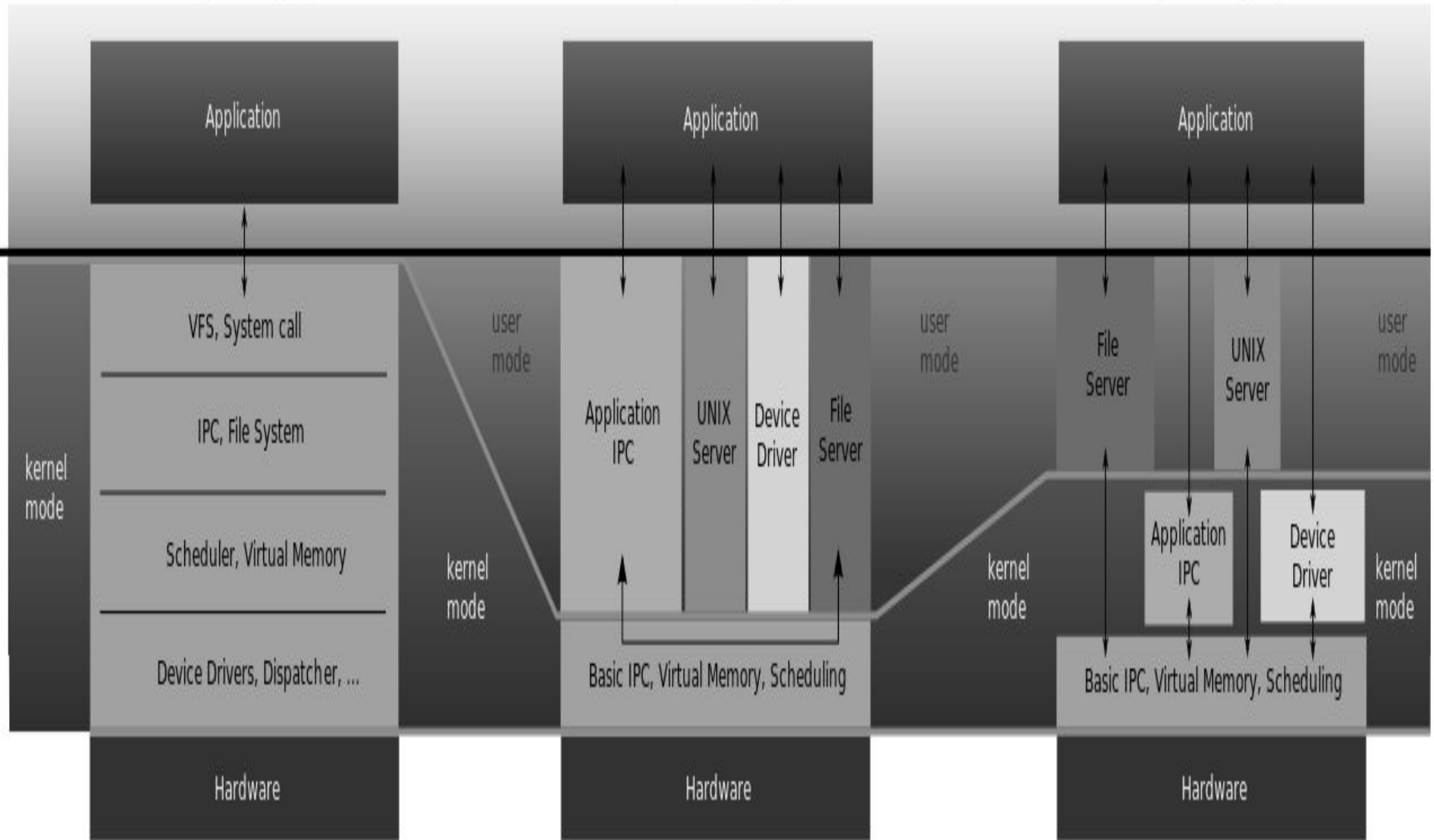
Monolithic Kernel based Operating System

Microkernel based Operating System

"Hybrid kernel" based Operating System

System

Operating system

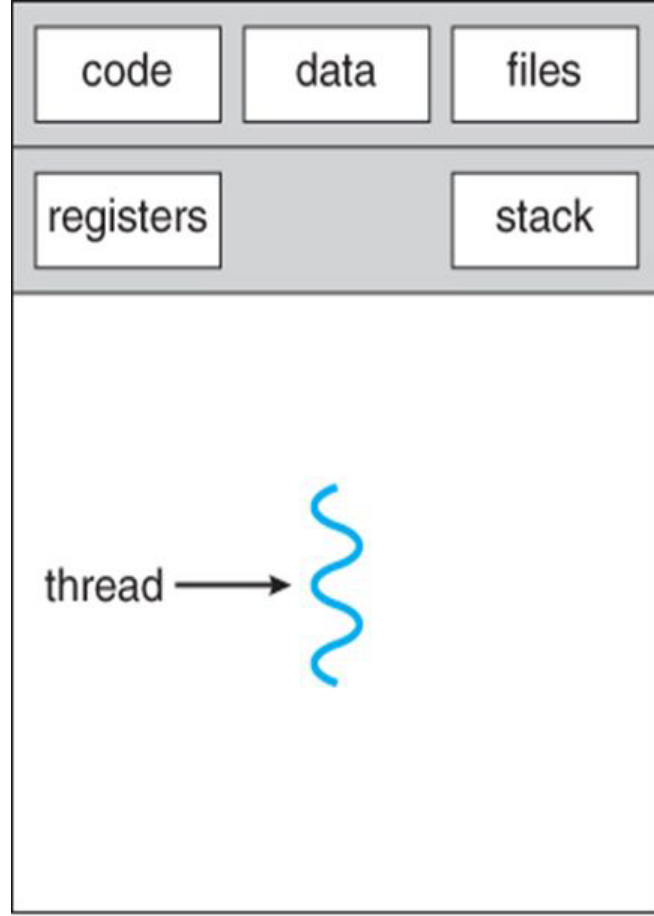


A process has a self contained execution environment that means it has a complete, private set of basic run time resources particularly each process has its own memory space. Threads exist within a process and every process has at least one thread

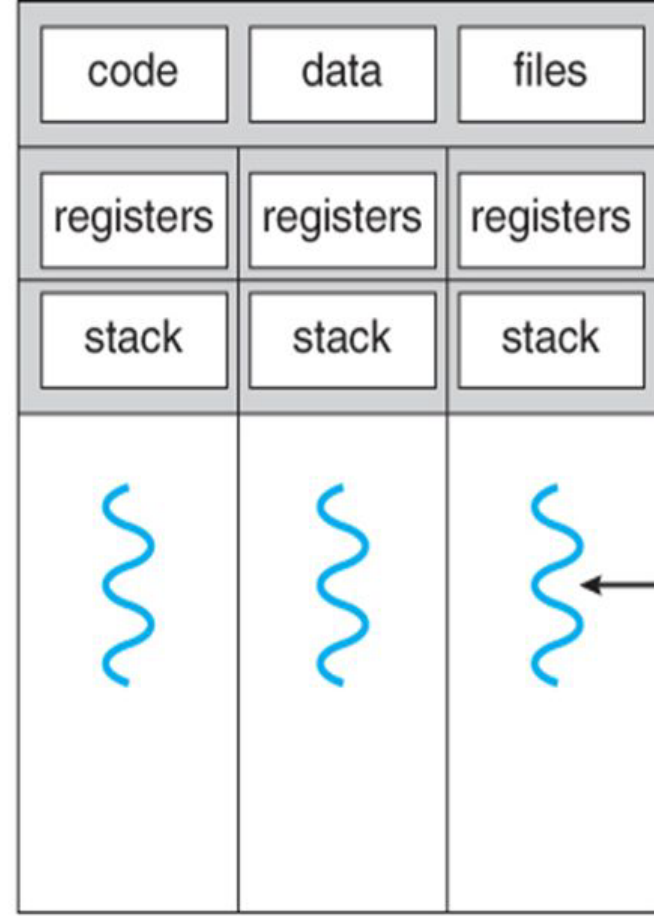
Each process provides the resources needed to execute a program. Each process is started with a single thread, known as the primary thread. A process can have multiple threads in addition to the primary thread.

On a multiprocessor system, multiple processes can be executed in parallel. Multiple threads of control can exploit the true parallelism possible on multiprocessor systems.

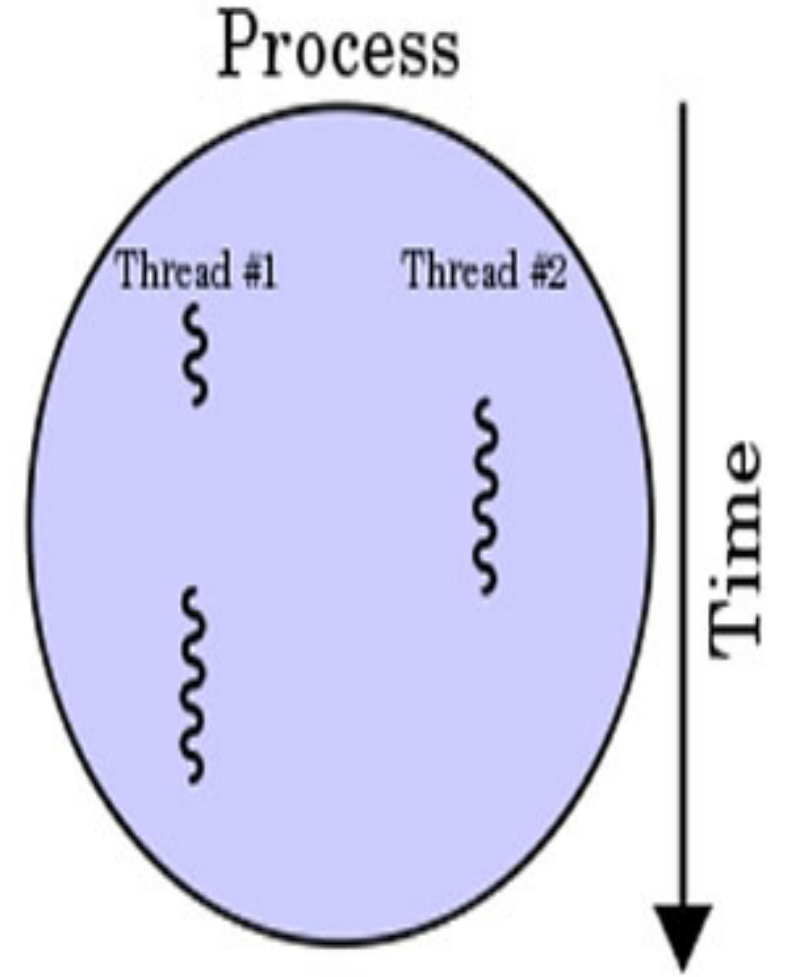
Processes are heavily dependent on system resources available while threads require minimal amounts of resource, so a process is considered as heavyweight while a thread is termed as a lightweight process.



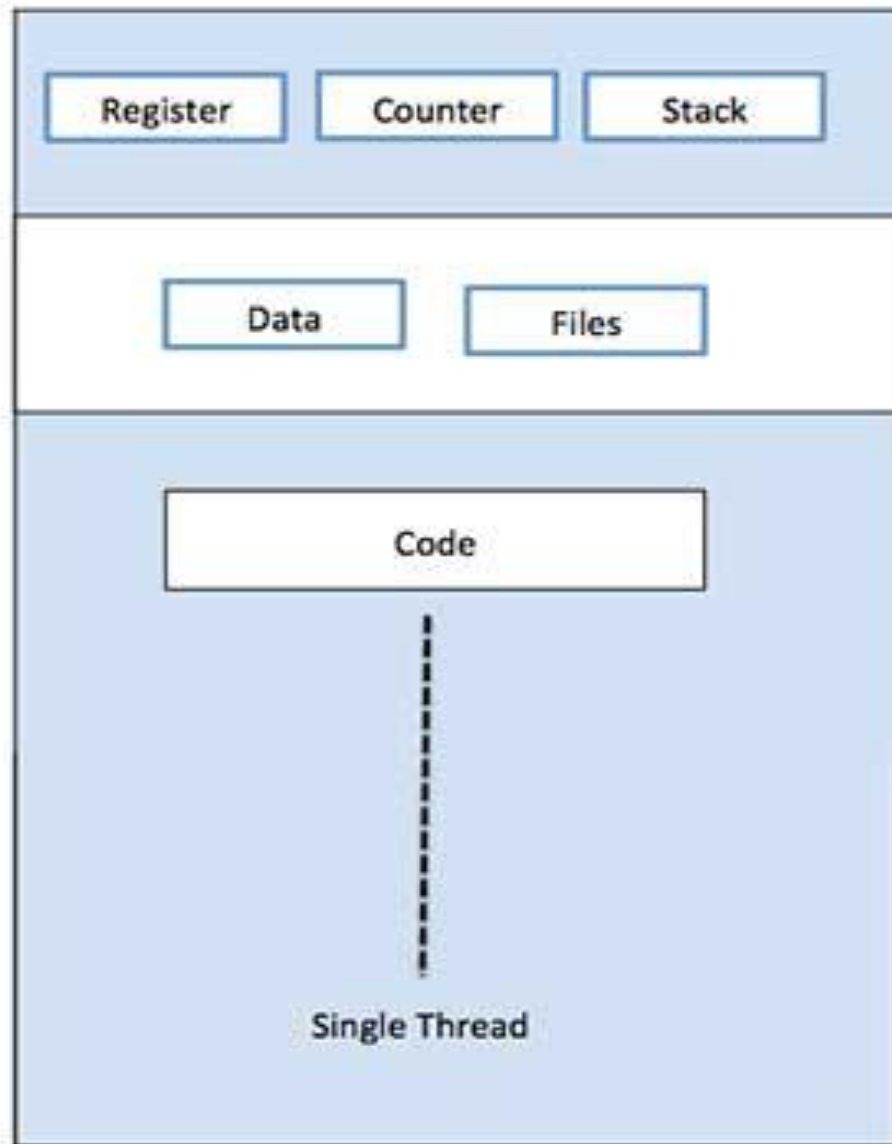
single-threaded process



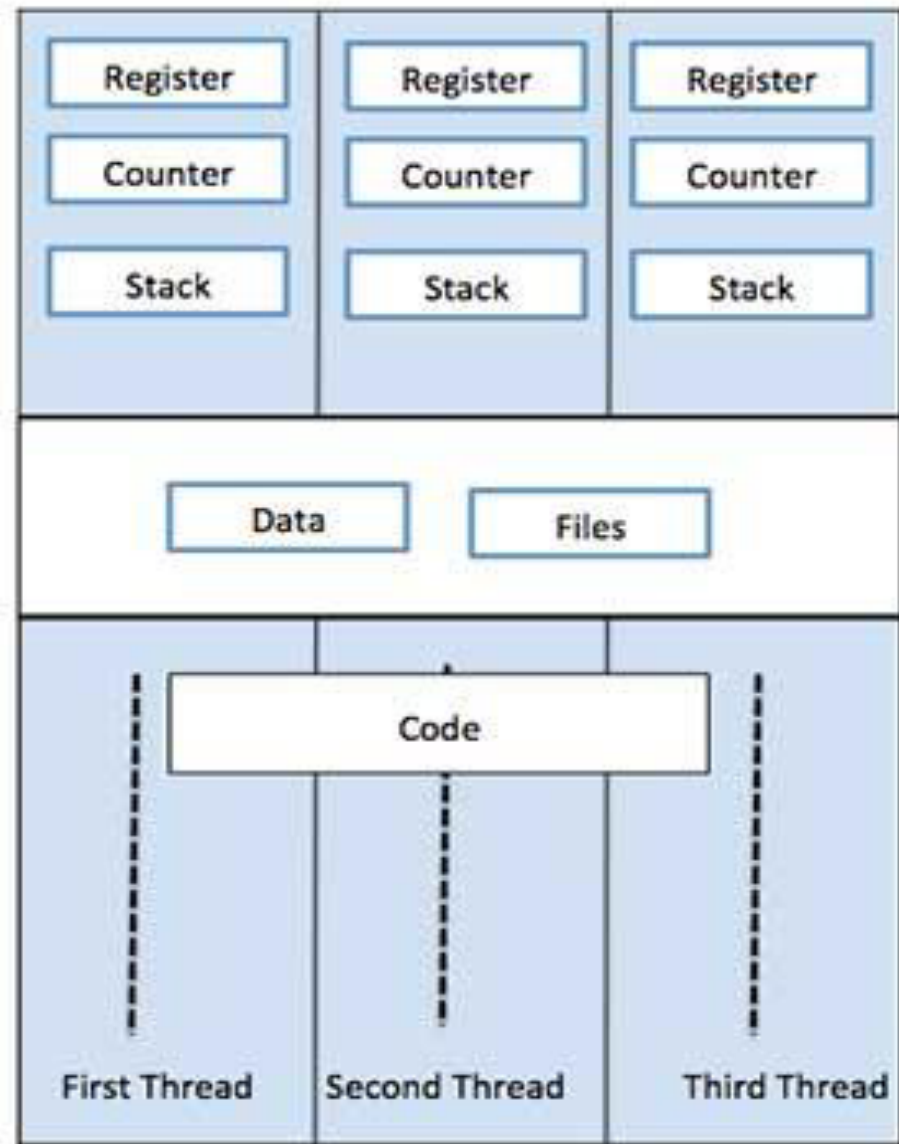
multithreaded process



- A thread is a basic unit of CPU utilization, consisting of a program counter, a stack, and a set of registers, (and a thread ID.)
- Traditional (heavyweight) processes have a single thread of control - There is one program counter, and one sequence of instructions that can be carried out at any given time.
- Multi-threaded applications have multiple threads within a single process, each having their own program counter, stack and set of registers, but sharing common code, data, and certain structures such as open files.



Single Process P with single thread



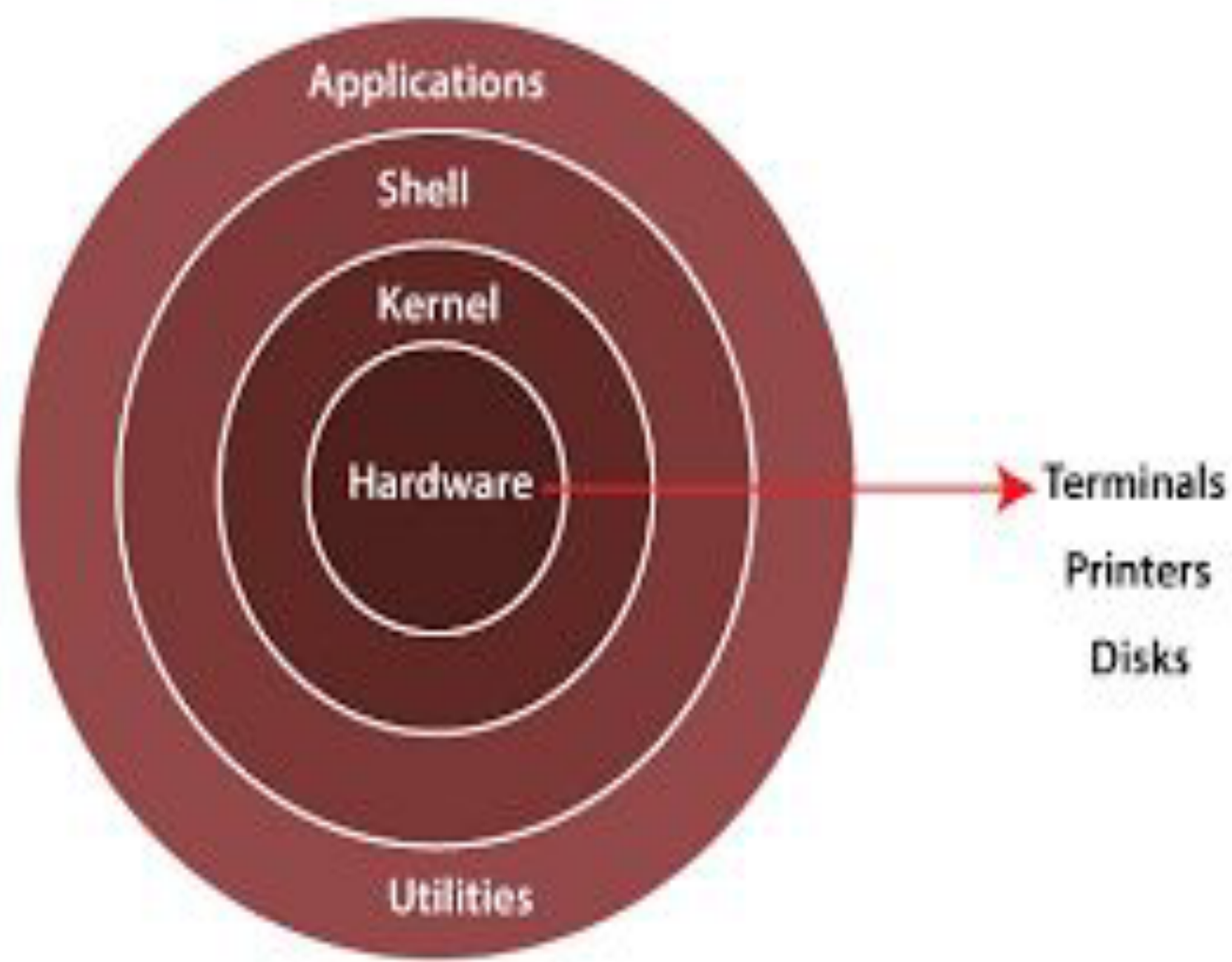
Single Process P with three threads

The shell (computer program) is the Linux command line interpreter. It provides an interface between the user and the kernel and executes programs called commands. Shells allow users to communicate efficiently and directly with their operating systems.

e.g. If a user enters [ls] then the shell executes the [ls] command.

Different Types of Shells in Linux

- The Bourne Shell (sh)
- The GNU Bourne-Again Shell (bash) - invented by Steven Bourne
- The C Shell (csh)
- The Korn Shell (ksh)
- The Z Shell (zsh)



The root directory denoted by the slash

An **Absolute Path** is a full path specifying the location of a file or directory from the root directory or start of the actual filesystem.

Example: /home/javatpoint/Desktop/CollegeStudent

The relative path of a file is its location relative to the current working directory. It never starts with a slash (/). It begins with the ongoing work directory.

Desktop/CollegeStudent

#!/bin/bash

A shell script to find the factorial of a number

read -p "Enter a number" num

fact = 1

for((i=2;i<=num;i++))

{
 fact=\$((fact*i))

}

echo \$fact

#!/bin/bash

A shell script to find the factorial of a number

read -p "Enter a number" num

fact=1

while [\$num -gt 1]

do

fact=\$((fact*num))

num=\$((num-1))

done

echo \$fact

Bash is a command-line interpreter or Unix Shell and it is widely used in GNU/Linux Operating System. It is written by Brian Jhan Fox.

The first line of our script file will be – `#!/bin/bash`

This will tell, the system to use Bash for execution. Then we can write our own scripts.

```
#!/bin/bash echo "Hello, GeeksforGeeks"
```

```
#!/bin/bash
```

```
Age = 17
```

```
if [ "$Age" -ge 18 ] ; then
```

```
    echo "You can vote"
```

```
else
```

```
    echo "You cannot vote" fi
```

To create and write a file with the .sh extension we can use [gedit](#) text editor.

The command for it will be – `gedit scriptname.sh`

Character	Meaning
~	Home directory
`	Command substitution (archaic)
#	Comment
\$	Variable expression
&	Background job
*	String wildcard
(Start subshell
)	End subshell
\	Quote next character
	Pipe
[Start character-set wildcard
]	End character-set wildcard

Character	Meaning
{	Start command block
}	End command block
;	Shell command separator
'	Strong quote
<">	Weak quote
<	Input redirect
>	Output redirect
/	Pathname directory separator
?	Single-character wildcard
!	Pipeline logical NOT

