

<b>Features</b>	<b>Multiprogramming</b>	<b>Multitasking</b>
<b>Basic</b>	It allows multiple programs to utilize the CPU simultaneously.	A supplementary of the multiprogramming system also allows for user interaction.
<b>Mechanism</b>	Based on the context switching mechanism.	Based on the time-sharing mechanism.
<b>Objective</b>	It is useful for reducing/decreasing	It is useful for running multiple processes at the

	CPU idle time and increasing throughput as much as possible.	same time, effectively increasing CPU and system throughput.
<b>Execution</b>	When one job or process completes its execution or switches to an I/O task in a multi-programmed system, the system momentarily suspends that process. It selects another process from the process scheduling pool (waiting queue) to run.	In a multiprocessing system, multiple processes can operate simultaneously by allocating the CPU for a fixed amount of time.
<b>CPU Switching</b>	In a multiuser environment, the CPU switches between programs/processes quickly.	In a single-user environment, the CPU switches between the processes of various programs.
<b>Timing</b>	It takes maximum time to execute the process.	It takes minimum time to execute the process.

<b>Parameters</b>	<b>Paging</b>	<b>Segmentation</b>
Individual Memory	In Paging, we break a process address space into blocks known as pages.	In the case of Segmentation, we break a process address space into blocks known as sections/segments.
Memory Size	The pages are blocks of fixed size.	The sections/segments are blocks of varying sizes.
Accountability	The OS divides the available memory into individual pages.	The compiler mainly calculates the size of individual segments, their actual address as well as virtual address.
Speed	This technique is comparatively much faster in accessing memory.	This technique is comparatively much slower in accessing memory than Paging.
Size	The available memory determines the individual page sizes.	The user determines the individual segment sizes.
Fragmentation	The Paging technique may underutilize some of the pages- thus	The Segmentation technique may not use some of the memory

	<b>Process</b>	<b>Thread</b>
Definition	A process is a program under execution i.e. an active program.	A thread is a lightweight process that can be managed independently by a scheduler.
Running mechanism	Processes run in separate memory spaces.	Threads within the same process run in a shared memory space.
Concept	Process is heavy weight and any program is in execution.	It is a lightweight process and a segment of a process.
PCB	The process has its own Process Control Block, Stack, and Address Space.	Thread has Parents' PCB, its own Thread Control Block, and Stack and common Address space.
Context Switching	Context switching between the process is more expensive	Context switching between threads of the same process is less expensive
Dependency	Processes are independent.	Threads are dependent
Controlling	Process is controlled by the operating system.	Threads are controlled by programmer in a program

<b>Sr. No</b>	<b>Long Term Scheduler</b>	<b>Short Term Scheduler</b>
1	It is job scheduler	It is CPU scheduler
2	It selects processes from job pool and loads them into memory for execution	It selects processes from ready queue which are ready to execute and allocates CPU to one of them
3	Access job pool and ready queue	Access ready queue and CPU
4	It executes much less frequently. It executes when memory has space to accommodate new process.	It executes frequently. It executes when CPU is available for allocation
5	Speed is less than short term scheduler	Speed is fast
6	It controls the degree of multiprogramming	It provides lesser control over degree of multiprogramming
7	It chooses a good process that is a mix-up of input/output bound and CPU bound.	It chooses a new process for a processor quite frequently.

<b>Parameter</b>	<b>Command Line Interface(CLI)</b>	<b>Graphic User Interface(GUI)</b>
<b>Definition</b>	Interaction is by typing commands	Interaction with devices is by graphics and visual components and icons
<b>Understanding</b>	Commands need to be memorized	Visual indicators and icons are easy to understand
<b>Memory</b>	Less memory is required for storage	More memory is required as visual components are involved.
<b>Working Speed</b>	Use of keyboard for commands makes CLI quicker.	Use of mouse for interaction makes it slow
<b>Resources used</b>	Only keyboard	Mouse and keyboard both can be used
<b>Accuracy</b>	High	Comparatively low
<b>Flexibility</b>	Command line interface does not change, remains same over time	Structure and design can change with updates



<b>Short Job First (SJF)</b>		<b>Shortest Remaining Time (SRTN)</b>	
It is a non-preemptive algorithm.		It is a preemptive algorithm.	
It involves less overhead than SRJF.		It involves more overheads than SJF.	
It leads to comparatively lower throughput.		It leads to increased throughput as execution time is less.	
It minimizes the average waiting time for each process.		It may or may not minimize the average waiting time for each process.	
It involves lesser number of context switching.		It involves higher number of context switching.	
Short processes are executed first and then followed by longer processes.		Shorter processes run fast and longer processes show poor response time.	
	Linux	Unix	
What is it?	Linux is an example of Open Source software development and Free Operating System (OS).	Unix is an operating system that is very popular in universities, companies, big enterprises etc.	
Cost	Linux can be freely distributed, downloaded freely, distributed through magazines, Books etc. There are priced versions for Linux also, but they are normally cheaper than Windows.	Different flavors of Unix have different cost structures according to vendors	
User	Everyone. From home users to developers and computer enthusiasts alike.	Unix operating systems were developed mainly for mainframes, servers and workstations except OSX, Which is designed for everyone. The Unix environment and the client-server program model were essential elements in the development of the Internet	

Manufacturer	Linux kernel is developed by the community. Linus Torvalds oversees things.	Three biggest distributions are Solaris (Oracle), AIX (IBM) & HP-UX Hewlett Packard. And Apple Makes OSX, an unix based os..
Usage	Linux can be installed on a wide variety of computer hardware, ranging from mobile phones, tablet computers and video game consoles, to mainframes and supercomputers.	The UNIX operating system is used in internet servers, workstations & PCs. Backbone of the majority of finance infrastructure and many 24x365 high availability solutions.
Development and Distribution	Linux is developed by Open Source development i.e. through sharing and collaboration of code and features through forums etc and it is distributed by various vendors.	Unix systems are divided into various other flavors, mostly developed by AT&T as well as various commercial vendors and non-profit organizations.
GUI	Linux typically provides two GUIs, KDE and Gnome. But there are millions of alternatives such as LXDE, Xfce, Unity, Mate, twm, ect.	Initially Unix was a command based OS, but later a GUI was created called Common Desktop Environment. Most distributions now ship with Gnome.
File system support	Ext2, Ext3, Ext4, Jfs, ReiserFS, Xfs, Btrfs, FAT, FAT32, NTFS	jfs, gpfs, hfs, hfs+, ufs, xfs, zfs format
Text mode interface	BASH (Bourne Again SHell) is the Linux default shell. It can support multiple command interpreters.	Originally the Bourne Shell. Now it's compatible with many others including BASH, Korn & C.
Price	Free but support is available for a price.	Some free for development use (Solaris) but support is available for a price.
Security	Linux has had about 60-100 viruses listed till date. None of them actively spreading nowadays.	A rough estimate of UNIX viruses is between 85 -120 viruses reported till date.



Paging	Segmentation
Paging divides the computer's primary memory into fixed-size units called page frames, and the program's address space into pages of the same size.	Segmentation is the only memory management technique that does not provide the user's program with a 'linear and contiguous address space.'.
The hardware memory management unit maps pages to frames.	Segments are areas of memory that usually correspond to a logical grouping of information such as a code procedure or a data array.
The physical memory can be allocated on a page basis while the address space appears contiguous.	Segments require hardware support in the form of a segment table which usually contains the physical address of the segment in memory, its size, and other data such as access protection bits and status.
Pages are used for swapping or managing memory.	Small pieces called segments are used for memory management.
Page is indicated by its number and offset.	Segment is indicated by segment number and its offset
Page table is formed	Segment table is formed.
Do not support user's view of memory.	Supports user's view of memory.

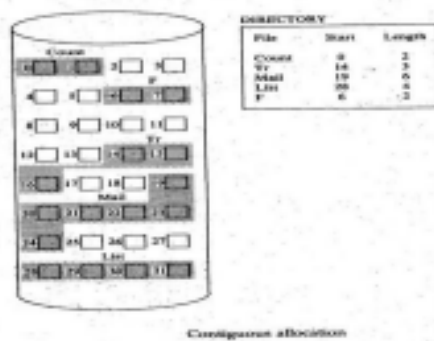
CRITERIA	LINUX	UNIX
<b>User interface</b>	Linux typically provides two GUIs, KDE and Gnome. But there are millions of alternatives such as LXDE, Xfce, Unity, Mate, twm, ect.	Initially Unix was a command based OS, but later a GUI was created called Common Desktop Environment. Most distributions now ship with Gnome.
<b>Name of Provider</b>	Redhat, Ubuntu, Fedora	Osx, Solaris, All LINUX
<b>Processing speed</b>	<b>Low:</b> As it is GUI based processing time is more as compare to UNIX.	<b>High:</b> As it is command based direct interpretation of commands is done so it takes less time as compare to LINUX.

<b>Security</b>	Linux has had about 60-100 viruses listed till date. None of them actively is spreading nowadays.	A rough estimate of UNIX viruses is <b>between</b> 85 -120 viruses reported till date.
-----------------	---	--

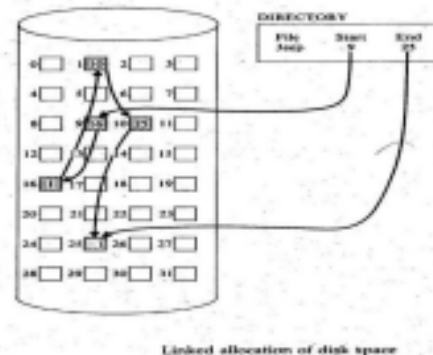
Sr No.	Multitasking	Multiprogramming
1	The process of executing multiple numbers of tasks simultaneously or concurrently is known as multitasking.	The process of executing multiple numbers of programs simultaneously or concurrently is known as multiprogramming.
2	Task is smallest unit of operation.	Program is larger unit of operation.
3	Tasks do not contain programs.	Program may contain multiple numbers of tasks.
4	User interaction is provided.	There is no user interaction to individual program.
5	In multitasking there are two types of labels, i.e. Foreground and Background.	In multiprogramming there are only programs.
6	It utilizes CPU efficiently.	It utilizes CPU as well as I/O devices efficiently.

Contiguous memory allocation	Linked memory allocation
Contiguous allocation requires that each file occupy a set of contiguous blocks on the disk.	Data structures are connected by a series of nodes.
Directory contains filename, starting block number and offset	Directory contains filename, pointer to starting block and ending block
Suffers from external fragmentation	There is no external fragmentation
Very difficult to find contiguous blocks of space	Any free blocks can be added to a chain.
Support direct access	Cannot support direct access
Size of file declared at start	Size of file need not be declared at start.
Insertions and deletions are difficult.	Insertions and deletions can be done easily.

Pointers not used.



Memory space required for pointer



Paging	Segmentation
It divides the physical memory into frames and program's address space into same size pages.	It divides the Computer's physical memory and program's address space into segments.
Page is always of fixed block size.	Segment is of variable size.
The size of the page is specified by the hardware.	The size of the segment is specified by the user.
It may lead to internal fragmentation as the page is of fixed block size.	It may lead to External fragmentation as the memory is filled with the variable sized blocks.
Page table is used to map pages with frames from memory.	Segment table is used to map segments with physical memory.
Page table contains page number and frame number.	Segment table contains segment number, length of segment and base address of segment from memory.
Invisible to Programmer	Visible to programmer
Paging consist of Static linking & dynamic loading	Segment consist of Dynamic Linking & Dynamic Loading
A page is of physical unit	A page is of logical unit

Pre-emptive Scheduling	Non Pre-emptive Scheduling
Even if CPU is allocated to one process, CPU can be preempted to other process if other process is having higher priority or some other fulfilling criteria.	Once the CPU has been allocated to a process the process keeps the CPU until it releases CPU either by terminating or by switching to waiting state.
Throughput is less	Throughput is high.
Only the processes having higher priority are scheduled.	Processes having any priority can get scheduled.
It doesn't treat all processes as equal.	It treats all process as equal
Algorithm design is complex.	Algorithm design is simple
Circumstances for preemptive (i) Process switch from running to ready state (ii) Process switch from waiting to ready state	Circumstances for Non-preemptive Process switches from running to waiting state Process terminates
For e.g.: Round Robin, Priority Algorithms	For e.g.: FCFS Algorithm

Monolithic OS	Microkernel OS
The entire O.S. is placed inside the kernel	Only bare minimum code is placed inside the kernel (only basic memory management and Inter Process Communication code)
It runs as a single large process	Here the kernel is broken down into processes called as servers
As all the services are placed inside the kernel, they have a single address space	As services (Servers provide services) are separated they have different address spaces
It is easy to implement/code	It is tough to implement/code
Performance is high (As kernel can invoke any function directly as everything is placed in the kernel)	Performance is low (As servers are separated, so to invoke services from other servers IPC(Inter Process Communication) is needed which requires kernel's permission and thus increases access time and lowers the performance)
Less Secure (If one service fails, entire system crashes)	More Secure (Even if one service crashes, others can function properly because of separation)

Explain critical section problem with example



<b>Sr. No.</b>	<b>Monolithic operating system structure</b>	<b>Microkernel operating system structure</b>
1	Kernel size is large	Kernel size is small
2	OS is complex to design	OS is easy to design, install and implement
3	Fast execution	Slow execution
4	All operating system services are included in kernel	Kernel provides only IPC and low level
5	No message passing, no context switching required while kernel is performing jobs.	It requires message passing and context switching
6	It is hard to extend	It is easy to extend