Date: .....
Page: .....

Name- Sandesh Poudel
Program- B.E software
Roll no : 19180075
Subject: Applied operating system.

Assignment - 1

Date: .....

Page: .....

1) Differentiate between multiprogramming, multitasking and multiprocess system.

Multiprogramming	Multitasking	Multiprocessing
1) The concurrent residency of more than one program in the main memory is called as multiprogramming.	1) The execution of more than one task simultaneously is called as multitasking.	1) The availability of more than one processor per system which can execute
2) No. of CPU: one	2) No. of CPU: ~~More than~~ one	2) More than one CPU
3) More time is taken to process the jobs.	3) Moderate amount of time.	3) less time is taken for job processing
4) One process is executed at a time.	4) one by one job is being executed at a time.	4) More than one process can be executed at a time.
5) One user at a time.	5) More than one user.	5) can be one or more than one users
6) Throughput is less.	6) Throughput is moderate.	6) Throughput is maximum.
7) Efficiency is less.	7) Efficiency is moderate.	7) Efficiency is maximum.
8) It is economical.	8) It is economical.	8) Is less economical.

Date: .....
Page: .....

2) Define Shell command and shell programming. Explain any ten DOS/ Linux Shell command with syntax and example. Write a simple DOS / Linux shell program to input two numbers and display the addition, subtraction, multiplication and division.

⇒ Shell commands are ways or instructions through which you can instruct your system to do some action.

Shell program is a program constructed of shell commands. They are interpreted each time they are run.

Shell commands

i) cat: It is used to display contents of input.
Example: cat hello.txt

ii) cp [old] [new]: It copies file.

iii) mv [old] [new]: It moves file or directory.
mv hello.txt v/bash.

iv) rm [path]: It removes a file.
rm hello.txt.

v) grep: select lines in files that match patterns.

vi) pwd: prints the user's current working directory.

vii) wc: counts lines, words and characters in its input.

viii) cd [path]: Changes current working directory.
example: cd. /folder / hello.

~~viii) pwd: prints the user's current working directory.~~

Date: .....

Page: .....

viii) ~~wc: counts~~

ix) ls: list the directories of current directory.

x) echo: print text to terminal window.

example echo "hello"

Program.

```
#!/bin/bash
sum=$((($1+$2))
subtract=$ (( $1+$2))
multiply=$ (($1+$2))
divide=$ (( $1+$2))

echo "sum is: $sum"
echo "subtract is: $ subtract"
echo "multiply is : $ multiply"
echo "Divide is : $ divide"
```

Output

```
./program.sh 6 3
sum is: 9
subtract is: 3
Multiply is: 18
Divide is : 2.
```

Date: .....
Page: .....

3) list the essential propecties for batch-orientation and interative operating system. For each of the following application which system (Batch or Interactive) is more suitable? state the reason.

a) word processing
b) Generating monthly bank statement
c) A flight simulator
d) Generating mark statement by univeesity.

⇒ Essential propecties of Batch orientation. It is particularly useful for operate that require the computer or peripheral device for an extended peciod of time and very little usee interaction.

- It is useful where we have to execute similar job together.

Interactive operating system

- provides the advantage of quick response.
- Avoids duplication of software.
- Reduce CPU idle time.

a) Word Processing

word processing refers to act of using Computer to create edit, save & print

Date: .....
Page: .....

documents. So, it will need some inputs or instructions from user so interactive OS is more suitable.

b) Generating monthly bank statements.

Generating monthly statement is a batch processing since the statement are generated in one go without one's ~~instruction~~. interaction.

c) A flight simulator

Flight simulator is a interactive device i.e it needs someone to control it so, it is a interactive process.

d) Generating mark statement by University.

Generating mark statement by university is a batch processing since the mark statements are generated in one go without one's interaction.

Date: .....
Page: .....

7) Write short notes on.

a) system implementation

Traditionally written in assembly language operating system can now be written in higher level languages. Code is written in high-level languages.

- can be written faster.
- Is more compact
- Is easier to understanding and ~~det~~ debug.

An operating system is far easier to port (move to same other hardware) if it is written in a high-level language.

b) System Generation.

System Generation is a utility that enables an operating system to configure hardware and software setups properly. Operating system are designed to run on any of a class of machines. the system must be configured for each specific computer site.

System generation program obtains information concerning the specific configuration of the hardware system.

- Booting: starting a computer by loading the kernel.

Date :
Page :

Bootstrap program code stored in ROM that is able to locate the kernel, load it into memory and start its execution.

Sandesh Poudel

Date: .....

Page: .....

Assignment-2.

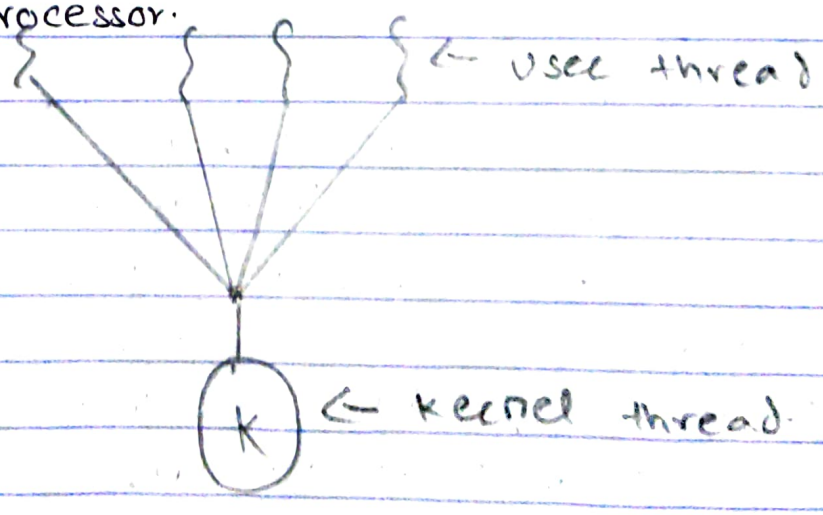
1) What is hyper threading? Explain different multithreading models with necessary diagrams.

=> Hyper-threading is a process by which a CPU divides up it's physical cores into virtual cores that are treated as if they are actually physical cores by the operating system.

We have three common types of threading models

a) Many-to-one Model:

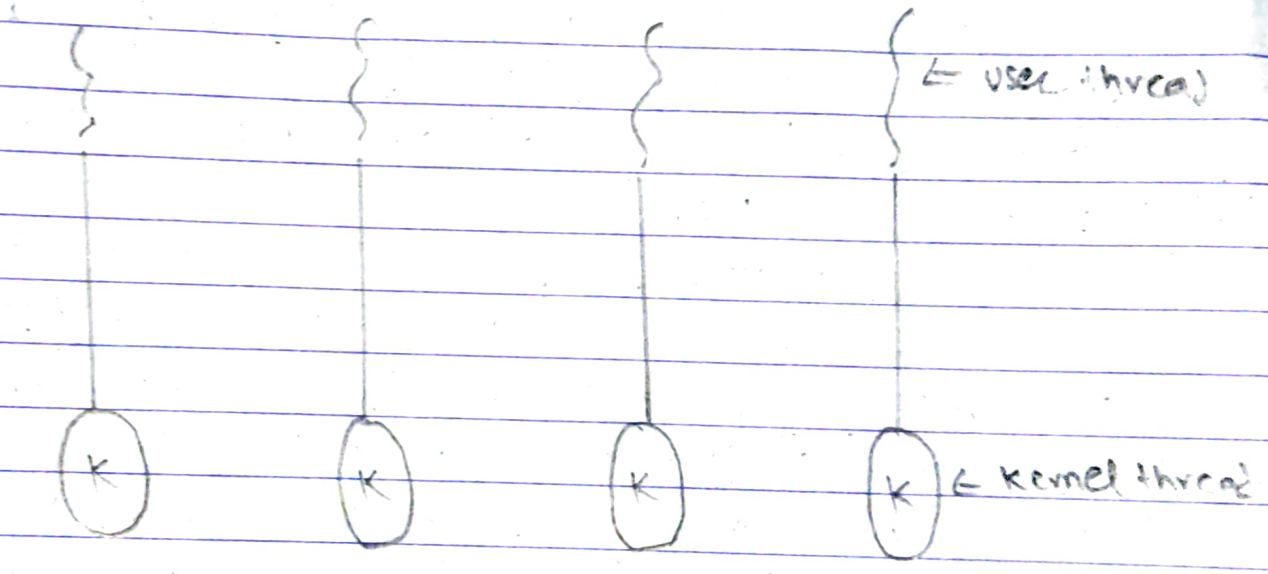
It maps ~~under~~ many user level threads to one kernel level thread Thread management is done in user space. So it is efficient; but the entire process will block if a thread makes a blocking system call. Also because only one thread can access the kernel at a time, multiple thread is unable to run in parallel on multiprocessor.



Date:
Page:

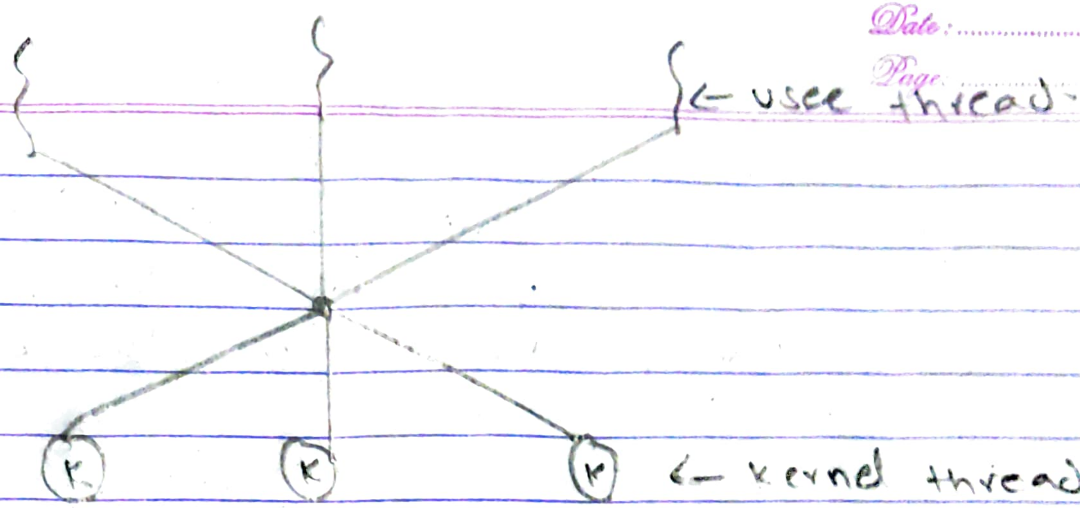
b) one-to one Model

It maps each user thread to kernel thread. It allows another thread to run when a thread makes a blocking system call. It also allows multiple thread to run in parallel in multiprocessor. The only drawback of this model is that creating a user thread requires creating the corresponding kernel thread.



c) Many-to-Many Model

It multiplexes many user-level threads to smaller or equal number of kernel thread. The number of kernel thread may be specific to either a particular application or a particular machine. This model is adopted by most of the operating system as it is over advantage of both previous model.



2) Define semaphore. List the characteristics, advantage and disadvantage of semaphore.

⇒ A semaphore is a synchronization tool, which is an integer variables that are used to solve the critical section problem by using two atomic operations, wait and signal that are used for process synchronization.

- wait

  The wait operation decrements the value of its argument s, if it is positive. If s is negative or zero, then no operation is performed.

```
wait (s) {
while (s<=0);
  s--; }
```

- Signal

  The signal operation increment the value of its argument s.

```
signal (s)
{ s++; }
```

Date:
Page:

Characteristics of semaphore.

1) Semaphore are like integer variable except that they have no negative value.
2) Only P and V operations is possible on semaphore.
3) The operation P and V are atomic.

Advantage of semaphore.

1) It allows only one process into the critical section.
2) They follow the mutal exclusion principle strictly and are much more efficient than some other synchronization.
3) There is no resource wastage because of busy waiting in semaphores as processor time is not wasted unnecessarily to check if the condition is fullfilled to allow process to access the critical section.
4) Semaphore are implemented in the machine independent code of the microkernel. So they are machine independent.

Disadvantages.

1) They are complicated so the wait and signal operations must be implemented in the correct order to prevent deadlocks.

Date: .....
Page .....

2) They are impractical for last scale use as their use leads to loss of modularity.

3) Semaphores may lead to a priority inversion where low priority process may access the critical section first and high priority process later.

5) Differentiate between deadlock and stravation giving some practical example.

Deadlock	stravation.
1) Deadlock happens when two or more process are waiting for each other.	1) stravation happens if same process is always choosen as victim.
2) A situation where two or more processes are unable to proceed because each is waiting for one of the other to do something.	2) It occurs if the waiting scheme for locked item is unfair, giving priority to some process over other.
3) It is also called circular waiting.	3) It is also called as lived lock.
4) It is a situation in which process are waiting for each other.	4) It means the process goes to a state where processes never progress.
5) There is no way to get out from deadlock, hence require some extend event to break deadlock.	5) There is a still chance that process may get out of lived lock however it is very low

Date:
Page:

6)	Process are waiting for an event that will never occur.	6) Process are indefinitely delayed because other process are given preference.
7)	Eg: Traffic.	7) Eg: In case of dining philosophers problem if among five philosophers only philosopher 2 and 4 always eat and other never get change to eat.

4) Explain 2PL what is the problem with this technique, Explain?

⇒ A transaction is said to follow the Two-Phase locking protocol if locking and unlocking can be done in two phases

1) Growing phase:
New locks on data items may be acquired but none can be released.

2) Shrinking phase.
Existing locks may be released but no new locks can be acquired.

	T_1	T_2
1)	lock-S(A)	
2)		lock-S(A)
3)	lock-S(B)	

Date: .....
Page: .....

1) W - - - - - -
5) Unlock (A)
6) Lock-X(C)
7) Unlock (B)
8) Unlock(A)
9) Unlock(C)
10) - - - - - - -

Transaction T_1

1) The growing phase (step 1-3)
2) shrinking phase (5-7)
3) lock point at 3.

Transation T_2

1) The growing phase (2-6)
2) The shrinking phase (8-9)
3) lock point at 6.

The problem with this technique are:

1) Seciality: It is guaranteed to happen.
2) ~~Cascading Rollback~~:

T_1	T_2	T_3
	Lock-S(A)	
		Lock-S(A)
	~~Lock-X(B)~~	
	Unlock(A)	~~Lock-X(C)~~
		~~Lock-X(C)~~

Date: .....
Page: .....

unlock(B)

lock-S(B)

unlock(A)

unlock(C)

~~lock-X(A)~~

unlock(B)

unlock(A)

so highlighted part are the last locks taken, i.e we can say the serial schedule could be $T_2 \rightarrow T_3 \rightarrow T_1$.

2) Cascading Rollback problem.

T_1	T_2	T_3
lock-X(A)		
R(A)		
W(A)		
lock-X(B)		
R(B)		
unlock(A)		
	lock(A)	
	R(A)	
	W(A)	
	unlock(A)	
		lock-S(A)
		R(A)

so if T_1 rollbacks then transaction T_2 & T_3 had

Date:
Page:

to rollback.

3) Deadlock Problem example.

T_1	T_2
lock-X(A)	
	lock-X(B)
lock-X(B)	
	lock-X(A)

So, T_1 is waiting for B and T_2 is waiting for A and both A and B are held by T_1 and T_2 respectively.

5) A system has three process and four allocable resources. The total four resource type exist in the amount as E = (4,2,3,1). The current allocation matrix and request matrix are as follow: Using Banker's algorithm, explain if this state is dead lock safe or unsafe.

Current Allocation Matrix

Process	R_0	R_1	R_2	R_3
P_0	0	0	1	0
P_1	2	0	0	1
P_2	0	1	2	0

Allocation Request Matrix

Process	R_0	R_1	R_2	R_3
P_0	2	0	0	1
P_1	1	0	1	0
P_2	2	1	0	0

Soln

E(Existing Resources) = (4, 2, 3, 1)

P(Processed Resources) = (2, 1, 3, 1)

A(Available Resources = E-P) = (2, 1, 0, 0)

Date:
Page:

- Process P_0 can't run to complete since Need > Available Resources.
- Process P_1 also can't run to complete since Need > Available Resources
- Now Process P_2 can complete since it's Need ≤ Available Resources.

when P_2 complete, it returns back all of it's resources then total available resources is,

$$A = (2,1,0,0) + (0,1,2,0) = (2,2,2,0)$$

- Again Process P_0 can't run to complete since Need > Available Resources.
- Now Process P_1 can complete since it's Need ≤ Available Resources.

when P_1 complete, it returns back all of it's resources then total available resource is

$$A = (2,2,2,0) + (2,0,0,1) = (4,2,2,1)$$

- finally Now Process P_0 can complete since it's Need ≤ Available Resources.

when P_0 run to completion, it return back all of its resources then total available resource is.

$$A = (4,2,2,1) + (0,0,1,0) = (4,2,3,1)$$

Hence, all the five process can run up to

completion in sequence P_2, P_1 and P_0. This implies that the system is safe and deadlock free.

6) Consider the following snapshot.

	Allocation				MAX				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P_0	0	0	1	2	0	0	1	2	1	5	2	0
P_1	1	0	0	0	1	7	5	0				
P_2	1	3	5	4	2	3	5	6				
P_3	0	6	3	2	0	6	5	2				
P_4	0	0	1	4	0	6	5	6				

Available - (1, 5, 2, 0)

Allocated - (2, 9, 10, 12)

Available = Existing - Allocated

Existing = (3, 14, 12, 12).

i) what is the content of need matrix.

ii) We know Need [i, j] = Max [i, j] - Allocation [i, j].

The need matrix is obtained as below.

	A	B	C	D
P_0	0	0	0	0
P_1	0	7	5	0
P_2	1	0	0	2
P_3	0	0	2	0
P_4	0	6	4	2

Date:
Page:

iv) Apply banker's Algorithm to explain, if the system is in safe state.

=) Applying safety algorithm, to determine whether the system is in safe state or not.

For P_i if Need $\leq$ Available, then P_i is kept in safe sequence.

- Now Process P_0 can complete since Need $\leq$ Available resources.

After P_0 completes, it returns all the resources.

Total A = (1, 5, 2, 0) + (0, 0, 1, 2)
= (1, 5, 3, 2)

- Process P_1 can't complete since Need > Available.
- Process P_2 can complete since Need $\leq$ Available.

After completion.

Total Resources (A) = (1, 5, 3, 2) + (1, 3, 5, 4)
= (2, 8, 8, 6)

- Process P_3 can complete since Need $\leq$ Available.

After completion,

Total Resources (A) = (2, 8, 8, 6) + (0, 6, 3, 2)
= (2, M, 11, 8)

Date: .....
Page: .....

- Process P_4 can complete since Need $\leq$ Available.
  After completion,
  Total Resources (A) = (2,14,11,8) + (0,0,1,4)
  = (2,14,12,12).

- Finally, Process P_1 can complete since Need $\leq$ Available
  After completion:
  A = (2,14,12,12) + (1,0,0,0)
  = (3,14,12,12).

Hence, all the five process can run up to completion in sequence P_0, P_2, P_3, P_4 and P_1. This simplies that the system is safe and deadlock free.

(iii) If a request from process P_1 arrives for (0,4,2,0) cand the request be granted immediately?

⇒ when request from process P_1 arrives for (0,4,2,0), then the reqvest can be currently assumed to be granted if Need $\leq$ Available. Now, new allocation become.

Allocation

	A	B	C	D
P_0				
P_1	0	0	1	0
P_2				
P_3				
P_4				

1 5 2 0

Date :

Page :

	Allocation.				Available			
	A	B	C	D	A	B	C	D
P_0	0	0	1	2	1	1	0	0
P_1	0	4	2	0				
P_2	1	3	5	4				
P_3	0	6	3	2				
P_4	0	0	1	4				

Need.

Process	A	B	C	D
P_0	0	0	0	0
P_1	1	3	3	0
P_2	1	0	0	2
P_3	0	0	2	0
P_4	0	6	4	2

Now, we must determine whether this new system state is safe. To do so, we again execute safety algorithm and find the sequence.

$<P_0, P_2, P_3, P_4, P_1>$ satisfies the safe requirement. Hence, we can immediately grand the request of process P_1.

Sandesh Poudel

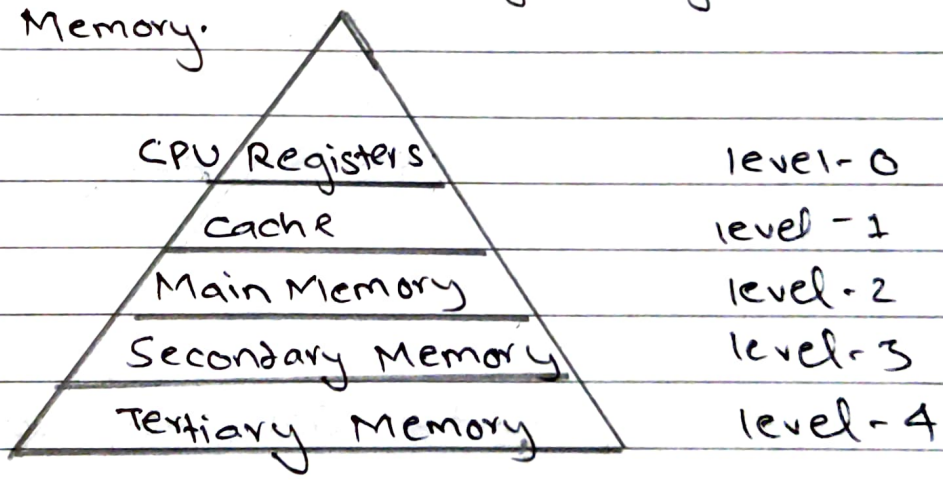
Date:
Page:

Assignment-3

1) what is memory hierarchy? Explain the need of memory hierarchy in detail.

=) The computer memory hierarchy is a pyramid structure which is used to describe the differences among memory types. It separates the computer storage based on hierarchy.

level 0: ~~Cache memory~~ CPU Registers
level 1: Cache memory
level 2: Main Memory
level 3: Magnetic disk or secondary memory
level 4: Optical disk or magnetic types or tertiary Memory.



In Memory Hierarchy the cost of memory, capacity is inversely proportional to speed.

Memory hierarchy is arranging different kinds of storage present on a computing device based on speed of access. At the very top, the highest performing storage is CPU registers which are the fastest to read and write to. Next is cache memory followed by

conventional DRAM memory, followed by disk storage with different levels of performance including SSD, optical and magnetic disk drives.

To bridge the processor memory performance gap, hardware designers are increasingly relying on memory at the top of the memory hierarchy to close/reduce the performance gap. This is done through increasingly larger cache hierarchies, reducing the dependency on main memory which is slower.

2) For the partion of 50k, 240k, 560k, 720k, 150k (in order), place the process of size 115k, 230k, 600k, 40k by using following concept.

a) Best fit.

Ans Given memory parition: 50k, 240k, 560k, 720k, 150k.

Process requiring spacing:
115k, 230k, 600k, 40k.

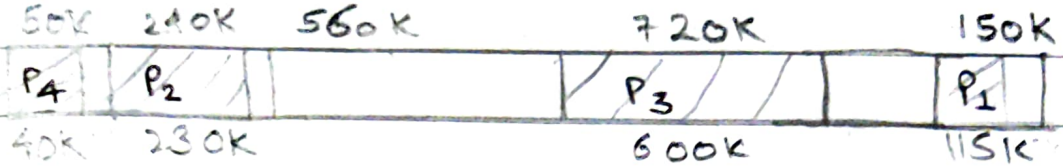
Let $P_1 = 115k$

$P_2 = 230k$

$P_3 = 600k$

$P_4 = 40k$.

a) Best Fit.



50k-40k = 10k (wastage)

240k-230k = 10k (wastage)

720k-600k = 120k (wastage)

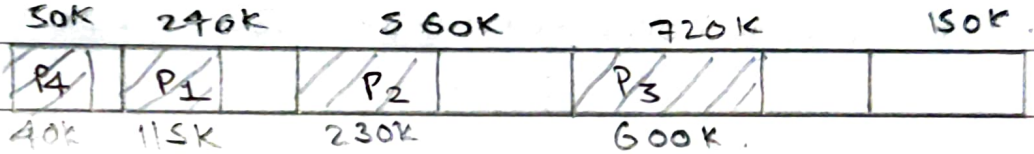
150k-115k = 35k (wastage)

Fragmentation = 10k + 10k + 560k + 120k + 35k

= 735k.

This algorithm chooses the free block to the requested size.

b) First Fit.



50k-40k = 10k (wastage)

240k-115k = 125k (wastage)

560k-230k = 330k (wastage)

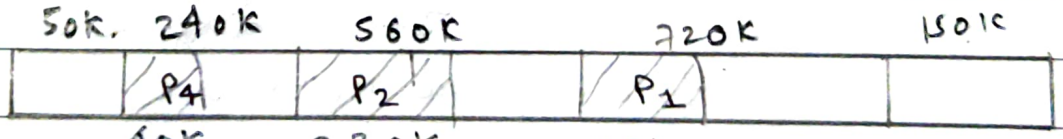
720k-600k = 120k. (wastage)

Fragmentation = 10k + 125k + 330k + 120k + 150k

= 735k.

This algorithm start searching for next memory ~~where it stop last time~~ from starting point

c) Worst Fit



(240-40)k = 200k.

560k-230k = 330k

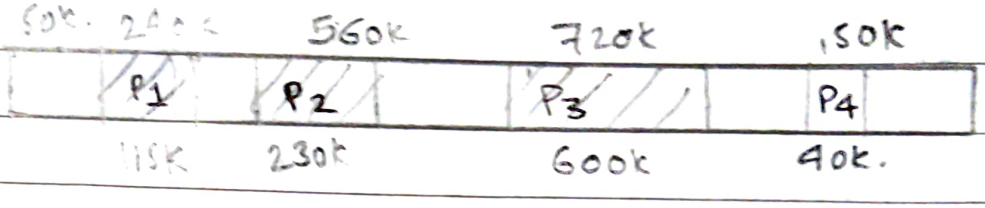
720k-115k = 605k.

∴ P_3 is not allocated in memory due to lack of space.

Fragmentation = 50k + 200k + 330k + 605k + 150k = 1335k.

Here, still 1335 K space is available but we cannot allow P_3 in memory because of worst fit.

iv) Next Fit.



240K-115K = 125K (wastage)
560K-230K = 330K (wastage)
720K-600K = 120K (wastage)
150K-40K = 110K. (wastage)

Fragmentation = 50K + 125K + 330K + 120K + 110K
= 735K.

3) Briefly explain multilevel page table.

=) Multiple Paging is a paging scheme which consist of two or more levels of page tables in a hierarchical manner. It is also known as hierarchical paging. The entries of the level 1 page table are pointers to level 2 page table and entries of the level 2 page table are pointers to a level 3 page table and so on. The entries of the last level page table are stores actual frame information. Level 1 contain single page

table and address of that table is stored in PTBR (page Table Base Register).

Virtual Address.

Level 1	Level 2	- - -	level n	offset

In multiple paging whatever may be level of paging all page tables will be stored in a main memory. So, it requires more than one memory access to get the physical address of page frame. once access for each level needed.

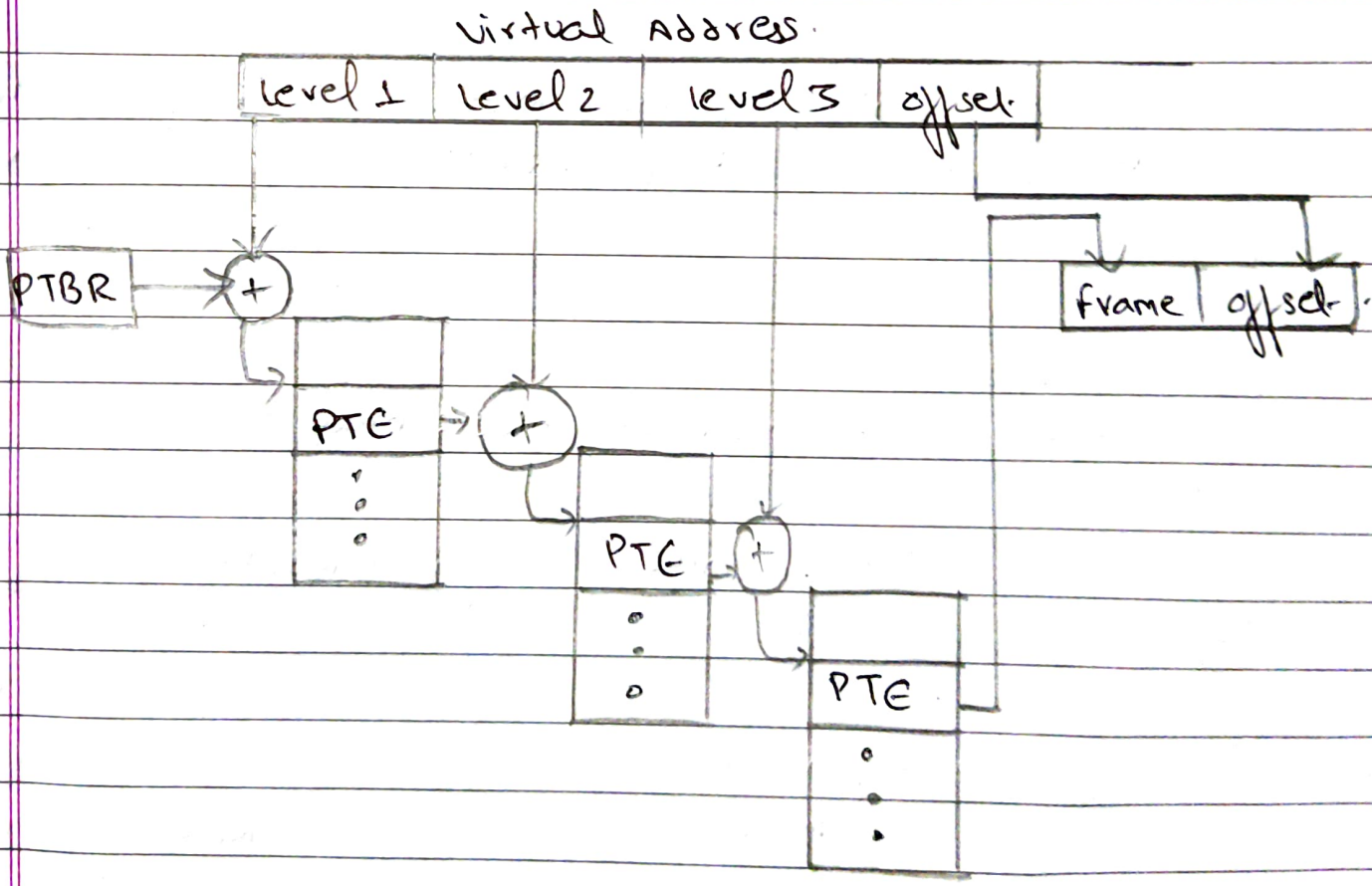


Fig: 3 level paging system.

Date.
Page.

4) Write short note on

i) Block Mapping.

In Block Mapping system represents address as ordered pairs. Block Mapping is a mapping of set of blocks onto the processes.

- Pages
  - Blocks are fixed size.
  - Technique is called paging
- Segments
  - Blocks may be different size
  - Technique is called segmentation

Virtual address format in a block mapping system.

Block number	Displacement	virtual address
b	d	$v = (b, d)$.

Block map origin register containing base address of process's block map table.

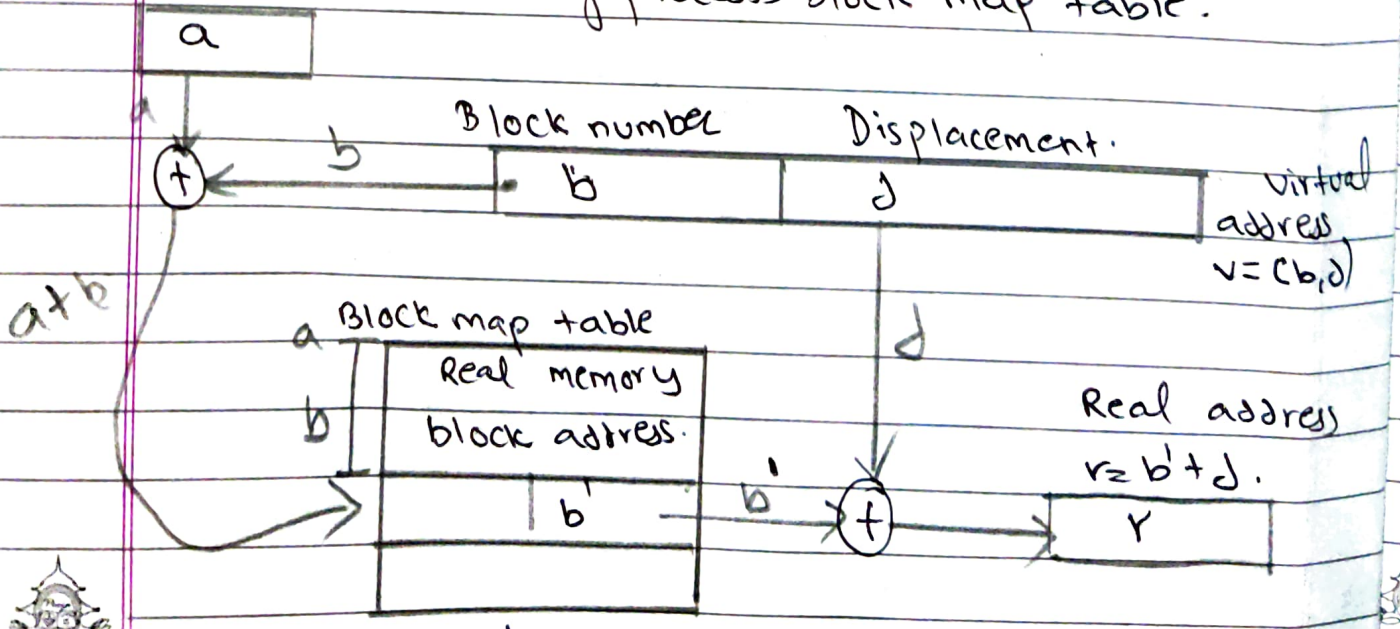
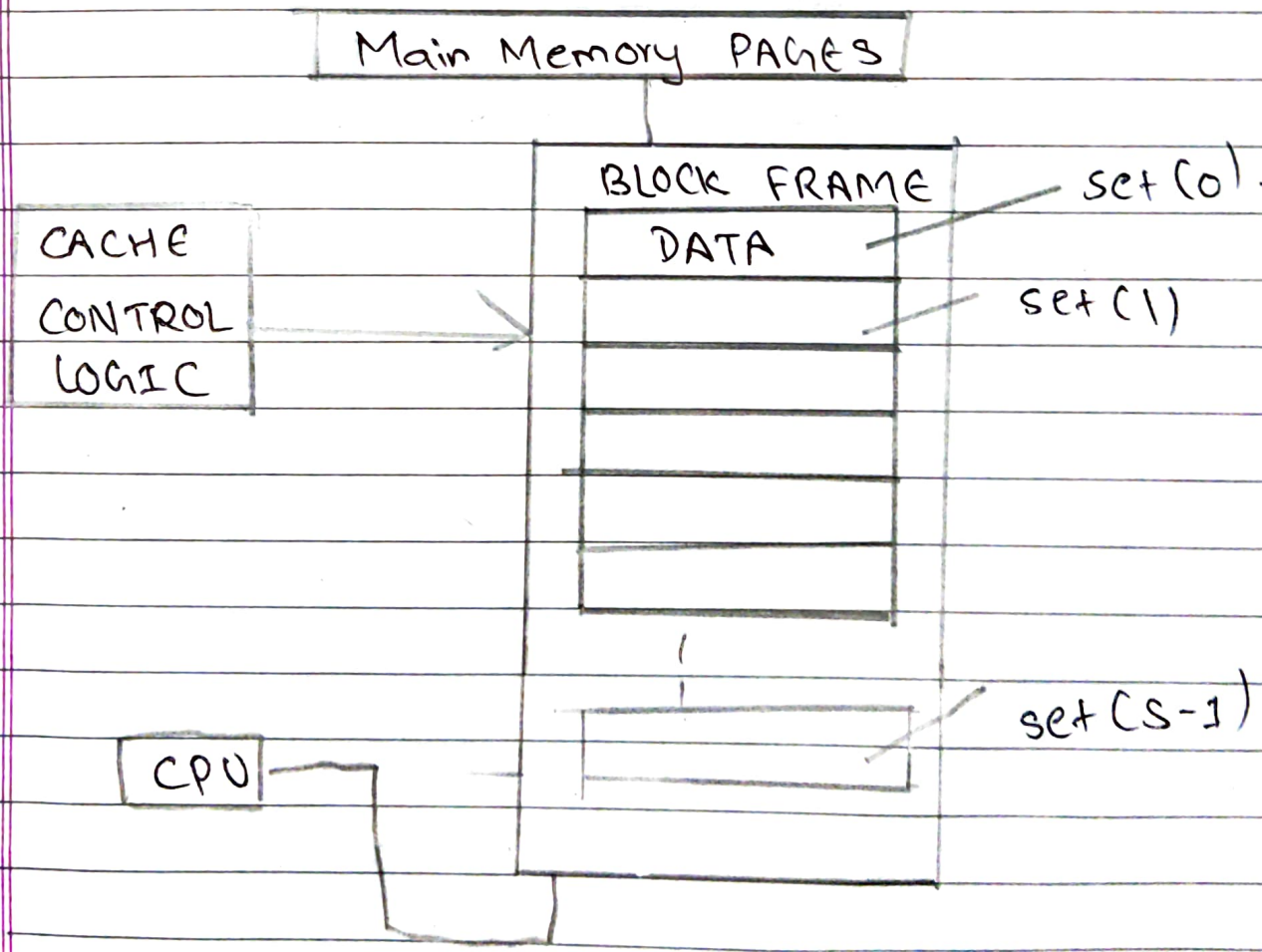


Fig: virtual address translation with block mapping

ii) Direct Mapping.

Direct Mapping is a very simplest mapping technique because in which every block of primary memory is mapped into with single possible cache line. In Direct mapping, every memory block is allocated for particular line in the cache memory. Some time memory block is engaged with recently cache line, then fresh block is required for loading and previously block is deleted. The address space is divided into two segments like as index field and tag field, and a tag field is saved into cache memory.



Sandesh Poudel

Date.
Page.

Assignment - 4

1) Difference between I/O Mapped vs Memory mapped I/O.

	Memory mapped I/O	I/O mapped I/O
1)	In this device address is 16-bit Thus A_0 and A_{15} lines are used to generate the device address.	1) In this device address is 8-bit. Thus A_0 to A_7 or A_{15} lines are used to generate device address.
2)	MEMR and MEMW control signals are used to control read and write I/O operations	2) IOR and IOW control signals are used to control read and write I/O operations
3)	Instruction available are LDA, STA, MOV, R.M, ADD M etc.	3) Instructions available are IN and OUT.
4)	Data transfer is between any register and I/O device	4) Data transfer is between accmulator and I/O device.
5)	Decoding 16-bit address may require more hardware	5) Decoding 8-bit address will require less hardware.

2) Differentiate between sector sparing vs sector splipping.

sector sparing	sector splipping
1) sector sparing uses spare sectors.	1) sector splipping doesn't uses spare sectors.
2) sector sparing results in copying of a single sector.	2) sector splipping may results in copying of multiple sectors.
3) sector sparing can help recover from hard errors.	3) sector splipping cannot recover from hard errors.
4) sector sparing per track to be used if another sector becomes defective.	4) sector splipping is a technique used to deal with defective sectors in hard disk drives

3) write short notes on

a) UNIX I/O kernel structure

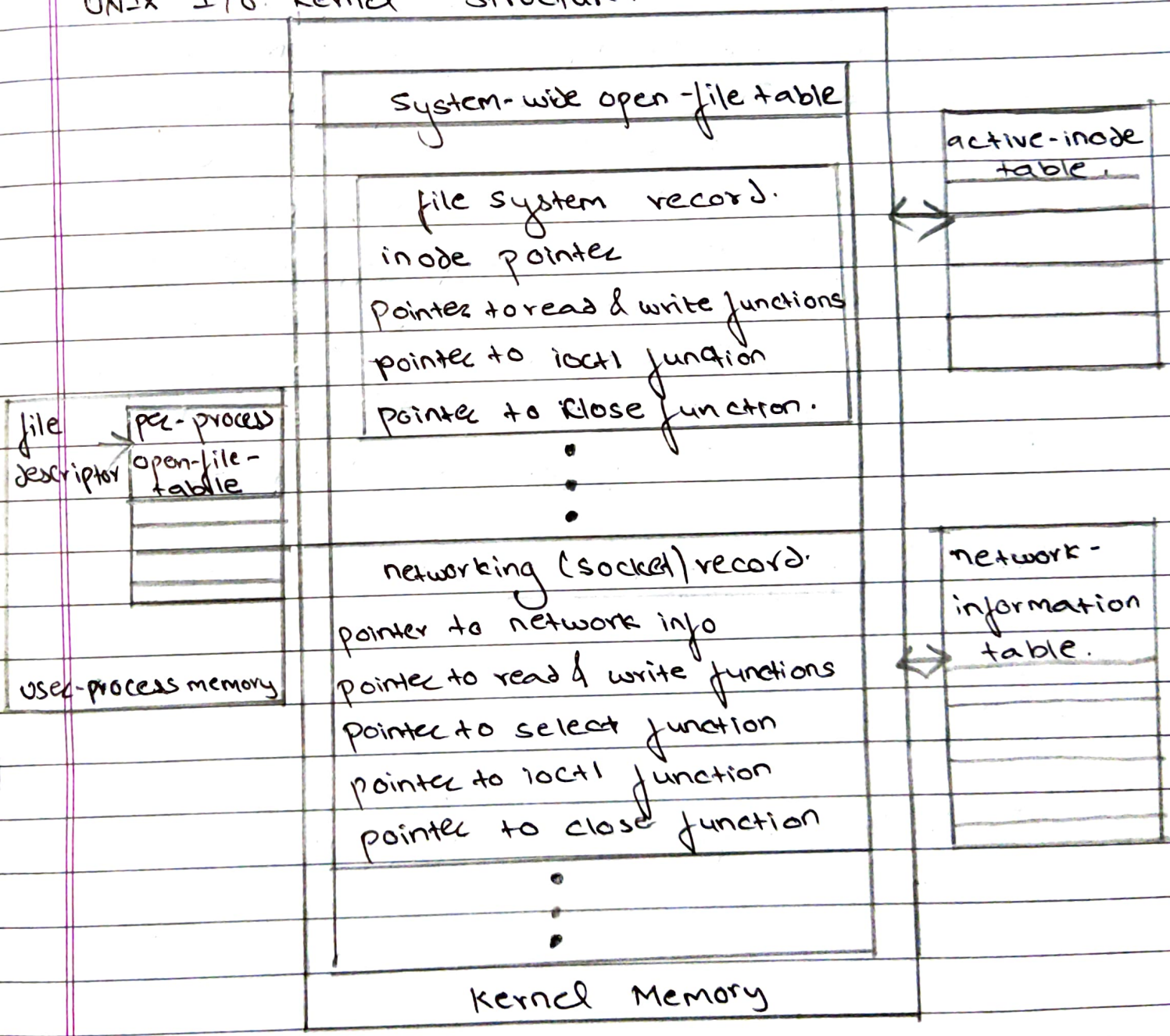
It is responsible for scheduling running of user and other processes. This architecture is responsible for allocating memory. It is responsible for managing the swapping between memory and disk. It receives service requests from the processes and honours them.

System Calls in UNIX

The following are typical systems call in UNIX: open a file open, close a file close, perform

I/O read and write, send a signal kill, terminate a process exit.

UNIX I/O kernel structure.



Date.
Page.

b) Storage Area Network

Storage Area Network is a dedicated, specialized and high speed network which provides block-level data storage. It delivers the shared pool of storage devices to more than one server. The main aim of SAN is to transfer the data between the server and storage devices. It allow for transferring the data between the storage systems. They are mainly used for accessing storage devices such as tape libraries and disk-based device from the servers. It is dedicated network which is not accessible through the LAN. Protocols of SAN are:

1) FCP (Fibre channel protocol)
2) iSCSI
3) FCoE
4) NVMe

It is better for disk utilization.

Sandesh Poude

Date.
Page.

Assignment-5

1) VIRUS and Worm

	VIRUS	WORM
1)	A virus is a malicious executable code attached to another executable file which can be harmless or can modify or delete data.	1) A worm is a form of malware that replicates itself and can spread to different computers via Network.
2)	The main objective of viruses to modify the information.	2) The main objective of worms is to eat the system resources
3)	It requires a host is needed for spreading.	3) It doesn't require a host to replicate from one computer to another.
4)	It is more harmful.	4) It is less harmful as compared
5)	Viruses can't be controlled by remote.	5) worms can be controlled by remote.
6)	It's spreading speed is slower	6) It's spreading speed is faster.
7)	Eg: Creeper, Blaster, slammer etc.	7) Eg: Morris worm, storm worm etc.
8)	It needs human action to replicate.	8) It doesn't need human action to replicate.

2) Capabilities.

Capability Based system which use capability lists to give protection to the resources. We can called resources as 'object' in protection system concept.

- Protection in capability Based system

Protecting the access of resources or objects by the program, processors and users present in the computer system. It protects the system from malicious and harmful viruses which are caused by programs, users and processors. In capability based protection system the access of the object are controlled by capabilities that's the reason it is named as capability based system.

- Types of capability Based system.

1) HYDRA
2) CAMBRIDGE CAP system

Hydra is a micro kernel which is anuclaw but not complete os. It provides a base on which different facilities of os is implemented. Eg: user can build their own file system on the top of hydra micro kernel os. As it provides security, it is used for experimentization.

Cambridge CAP system is simple and less powerful than hydra. It is divided into two

parts

1) Data capability.
It provide access to object but only basic operation are performed.

2) Software capability.
It is not interpreted by the OS. But protection is given to both cambridge cap system.

3) User Authentication and Authorization.

Authentication is act of validating that users are whom they claim to be. This is the first step in any security process. Complete an authetication process with:

- password: usernames and passwords are the most common authentication factors.
- One-time pins: Grant access for only one session or transaction
- Authentication apps: Generate security codes via an outside party that grants access.
- Biometrics: A user presents a fingerprint or eye scan to gain access to the system.

Authorization is system security is the process of giving the user permission to access a specific resource or function.

Giving someone permission to download a

Date.
Page.

Particular file on a server or providing individual users with adminstrative access to an application are good example of authorization.

In secure environments, authorization must always follow authentication.

Authentication	Authorization
confirms users are who they say they are.	Gives user permission to access a resource.