26.03.21

IAT-I

CS8493 Operating System

Jayashree. M
19CSE005
II-CSE-A

1)

Tightly coupled system	Loosely coupled system.
→ Tightly coupled system have a shared memory concept	→ loosely coupled system have a disturbed memory concept.
→ The data rate of tightly coupled system is high	→ The data rate of loosely coupled system is low
→ It is more expensive but compact in size.	→ It is less expensive but larger in size
→ Power consumption is low	→ power consumption is high
→ Security is high	→ security is low

2) Objective of operating system:

To provide an environment for a computer user to execute programs on computer hardware in a convenient and efficient manner.

To allocate the separate resources of the computer as needed to solve the problem given. The allocation process should be as fair and efficient as possible.

As a control program it serves two major functions.

i) supervision of the execution of user programs to prevent errors and improper use of the computer, and

ii) Management of the operation and control of I/o devices.

3) Operating systems are designed to run on any of a class of machine at a variety of sites with a variety of peripheral configurations. The system must then be configured or generated for each specific computer site, a process known as system generation SYSGEN

After an operating system is generated, it must be made available for use by the hardware. The procedure of starting a computer by loading the kernel is known as booting the system. A small piece of code known as the bootstrap program or bootstrap loader locates the kernel, loads it into main memory, and starts its execution.

4) The system calls acts as a interface to a running program and the operating system. These system calls available in assembly language instruction

1) Process control 2) File management 3) Device management 4) Information maintenance 5) Communication.

5) In multiprocessor systems, failure of one processor will not halt the system, but only slow it down. If there are ten processors & if one fails the remaining nine processors pick up the work of the failed processor. This ability to continue providing service is proportional to the surviving hardware is called graceful degradation.

6) The time taken by the dispatcher to stop one process and start another running is known as dispatch latency.

7) An I/o bounded process is one that spends more of its time doing I/o that it spends doing computations.

A cpu bound process, in contrast, generates I/o requests infrequently, using more of its time doing computations.

8) Any kind of sequential program is not a good candidate to be threaded. An example of this is a program that calculates an individual tax return.

Another example is "shell" program such as the C-shell or korn shell. Such a program must closely monitor its own working space such as open files, environment variables, and current working directory.

Part-B:

9) a) → Caches are useful because they can increase the speed of the average memory accen and they do so without taking up as much physical space as the memory hierarchy's lower elements do

→ When we add a cache, it would also bring economic and space Penalty and also an additional level of complexity

→ when we make a cache a large as a disk it would not be that effective because it would be very costly the huge size would slow it down

→ cache is a volatile memory, but on We want data to be persistent.

→ The Cache memory is invisible to $, it interacts with other memory management hardware

Motivation:

This limitation has been a significant Problem because of the persistant mismatch

The solution is to provide the small fast memory cycle time, comparable to processor cycle times this is very costly

Cache Principle

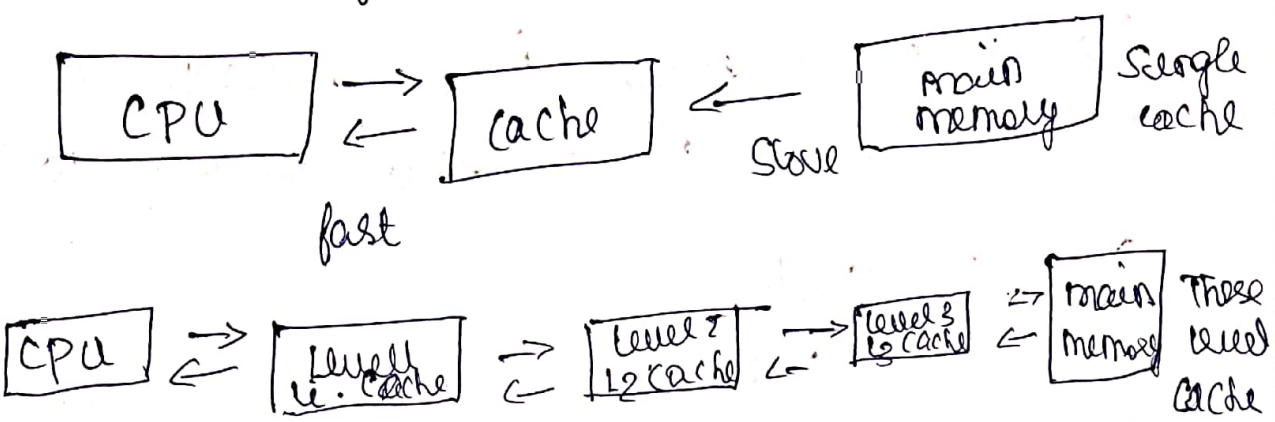
There is a large slower, main memory present with the smaller faster, Cache memory

Cache contains a copy of portion of main memory

Here when Processor wants to read a byte / words form memory, Initially it is checked if the byte / words is cache

If it is present, byte / word is transferred to processor

If not present, a block of main memory it read into cache and the checked for the byte / word.



→ Here L_2 is slower and large than L_1

→ L_3 is slower and large than L_2

⇒ main memory consist of upto 2^n addressable words, with each word having unique n bit address,

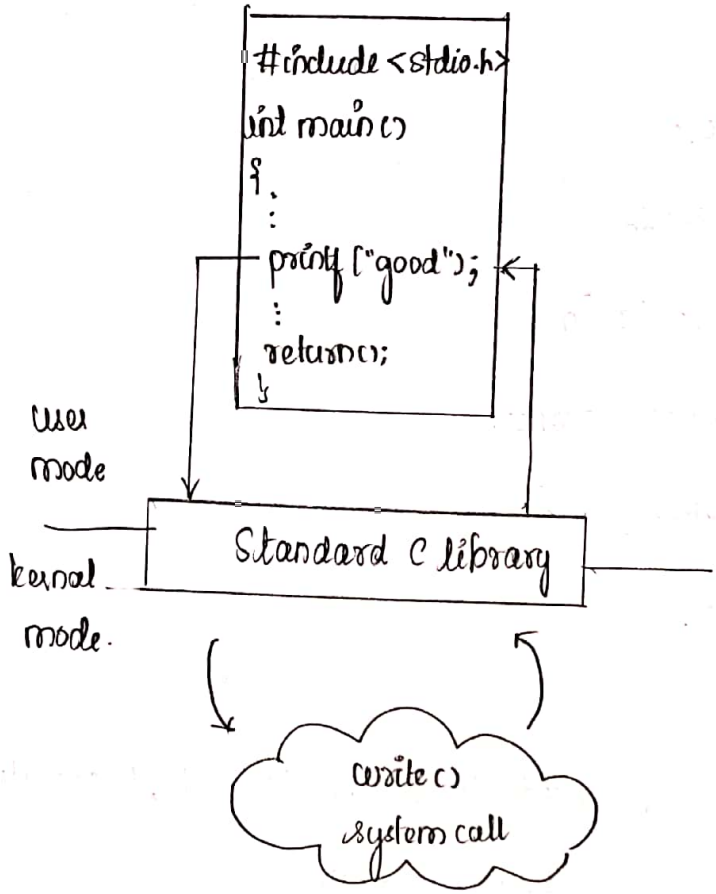
⇒ cache memory consists of a number of fixed blocks of k words each.

10) a) Types of system call:

* Process control
* File manipulation
* Device manipulation
* Information maintenance.
* Communication.
* Protection.

Process control:

* A running program needs to be able to halt its execution either normally (end()) or abnormally (abort())
* The system calls associated with process control includes
  * end, abort
  * load, execute
  * create process, terminate process
  * get process attributes, set process attributes
  * wait for time.
  * wait event, signal event
  * allocate & free memory.



File management:-

* File management
* create file, delete file
* open, close
* read, write, reposition
* get file attributes, set file attributes.

In order to work with files we first need to be able to create() and delete() files.

We may also need open(), read(), write(), recomposition

Device management:

* request, device, release device.
* read, write, reposition
* get device attributes, set device attributes.
* logically attach or detach devices.

A process may need several resources to execute - main memory, disk drives, access to file and so on. If the resources are available, they can be granted and control can be returned to the user process.

Information Maintenence:

* get time or date, set time or date
* get system data, set system data.
* get process, file, or device attributes
* set process, file, or device attributes.

* Many system call exist simply for the purpose of transferring information b/w the user program and the OS.

Communication:-

* Create, delete communication connection
* Send, receive messages
* Transfer status information
* attach or detach remote devices.

There are two common models of interprocess communication the message passing model and the shared-memory model.

The source of the communication known as the client, Receiving daemon known as server.

Protection:-

* protection provides a mechanism for controlling access to the resources provided by a computer system.
* system call providing protection includes set permission() and get permission(), which manipulate the permission settings.
* The allow user() and deny user() system call specify whether particular user can-or cannot be allowed to access certain resources.

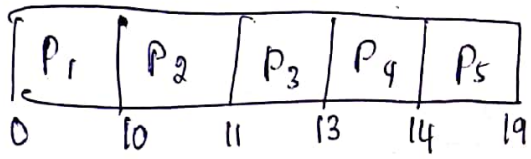
Part-c:

11. a)

	BT	Prior
P_1	10	3
P_2	1	1
P_3	2	3
P_4	1	4
P_5	5	2

process are arrived at P_1, P_2, P_3, P_4, P_5 order at time 0.

i)

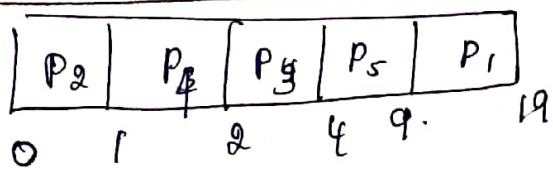


Process	ET	WT	TAT
P_1	10	0	10
P_2	1	10	11
P_3	2	11	13
P_4	1	13	14
P_5	5	14	19
		48	67

total waiting time = 48 ms

total turn around time } = 67 ms.

ii)

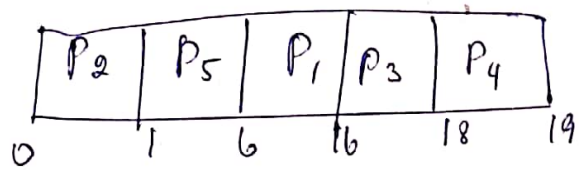


Process	ET	WT	TAT
P2	1	0	1
P4	1	1	2
P3	2	2	4
P5	5	4	9
P1	10	9	19
		16ms	35ms

total waiting time = 16ms

total turn around time } = 35ms

iii)

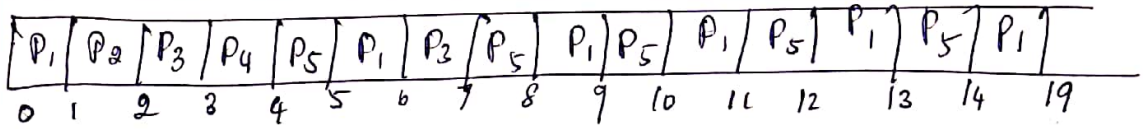


process	ET	prior	WT	TAT
P2	10	3	0	1
P5	1	1	1	6
P1	2	3	6	16
P3	1	4	16	18
P4	5	2	18	19
			31ms	60ms

total time around time } = 60ms

total waiting time } = 31ms

iv)



Process	ET	WT	TAT
P_1	10	9	19
P_2	1	1	2
P_3	2	5	7
P_4	1	3	4
P_5	5	9	14
		27 ms	46 ms

total waiting time = 27ms

total turn around time = 46ms