Unit test-01

OS(18CS43)

Shahul Hameed.S
1KN18CS097
CSE 'A' Sec
4th Sem

(01) What is operating System? Explain abstract view of components of Computer Systems.

An OS is an intermediatory b/w the user of the computer and the computer hardware. It provides a basis for application program and acts as an intermediary b/w user of computer and computer hardware.

Abstract / System view :-

we can view system as resource allocator i.e a computer System has many resources that may be used to solve a problems The OS acts as a manager of these resources. The OS must decide how to allocate these resources to program and the users So that it can operate the computer system efficiently and fairly. A different view of an OS is that it need to control various I/O devices and user programs i.e an OS is a control program used to manage the execution of user program to prevent errors and improper use of the computer.

Resources can be either CPU Time, memory Space, file storage Space, I/O devices and so on. The OS must support the following tasks.

a. provides the facility to create, modification of programs and data files using an editors.

b. Access to compilers for translating the user program from high level to machine language.

c. provide a loader program to move the compiled program code to computer memory for execution.

d. provides routines that handle the details of I/O programming.

(2) Compare and contrast multiprogramming and time sharing systems.

Time Sharing	Multiprogramming
* Time Sharing is the logical extension of multiprogramming, in this time sharing OS many users / processes are allocated with computer resources in respective time slots.	* Multiprogramming OS allows to execute multiple processes by monitoring their process states and switching in b/w processes.
* processor time is shared with multiple users thats why it is called as time sharing OS.	* processor and memory underutilization problem is resolved and multiple programs runs on CPU that's why it is multiprogramming.
* two or more users can use a processor in their terminal.	* processor can be executed by a single processor.
* Time Sharing OS has fixed time slice	* Multi-programming OS has no fixed time slice.
* Here the system works for the same or less time on each processes.	* Here the system does not take same time to work on different processes.
* System model of time sharing system is multiple programs and multiple users.	* System model of multi programming system is multiple programs.
Eg: Windows NT	Eg: Mac OS.

Unit Test-02

OS (18CS43)

Shahul Hameed.S
1FN18CS097
CSE A' Sec
4th Sem

① What is inter process communication? Explain direct and indirect IPC?

Mechanism for processes to communicate and to synchronize their actions. Message system - processes communicate with each other without resorting to shared variables. IPC facility provides two operations. Send and receive.

Direct Communication:-

Processes must name each other explicitly.

Links are established automatically. A link is associated with exactly one pair of communicating processes. B/W each pair there exist exactly one link. The link may be unidirectional, but is usually bi-directional.

Indirect Communication:-

Messages are directed and received from mail boxes. Each mail box has a unique id. Processes can communicate only if they share a mailbox.

Links established only if processes share a common mailbox

A link may be associated with many processes.

operations create a new mailbox, send and receive messages through mailbox and destroy mailbox.

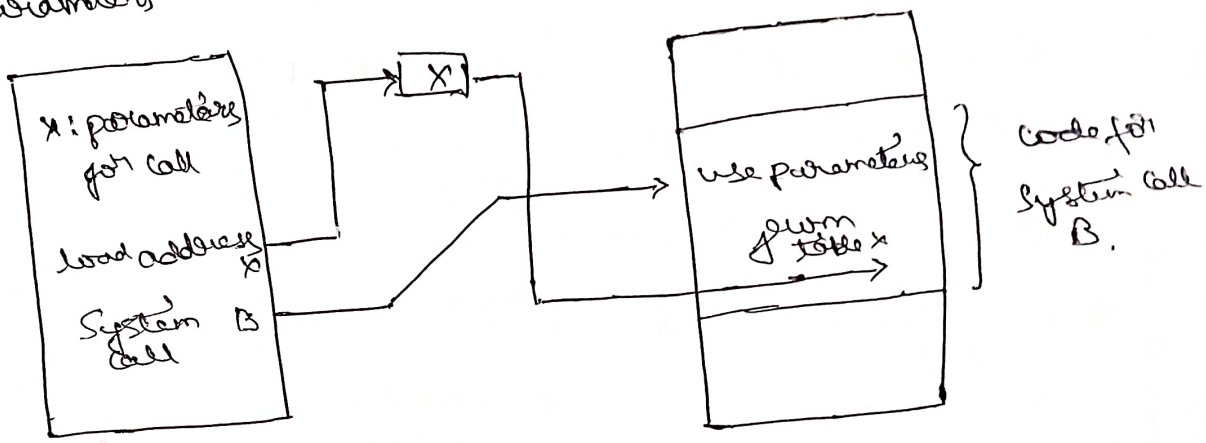
primitive are defined as: send(A, message) - send a message to mailbox A receive (A, message) - receive a message from mailbox A.

(2) What are system calls? Briefly explain/point out types of system calls.

> System provides interface b/w the process and OS. The calls are generally available as assembly language instruction and certain system allow system calls to be made directly from a high level language program.

Types of system calls:-

Passing parameters to OS



These general methods are used to pass the parameters to the OS.

The simplest approach is to pass the parameters in registers.

In some there can be more parameters than registers. In generally in a block or table in memory and the address of the block is passed as parameters in register. This approach used by linux.

* System calls may be grouped roughly into 5 categories.

→ process control.

→ File management.

→ Device management

→ Information maintenance

→ Communication.

unit test - 03

OS(18CS43)

Shahul Hameed. S
1KN18CS097
CSE 'A' Sec
4th Sem

(1) Deadlock is a situation where a set of processes are blocked because each process is holding a resources and waiting for another resource held by some other process.

Necessary conditions:

There are 4 conditions that are necessary to achieve deadlock.

(1) Mutual Exclusion:

* At least one resource must be held in non-sharable mode.
* If any other process requests this resource, then the requesting-process must wait for the resource to be released.

(2) Hold and wait:

* A process must be simultaneously → holding atleast one resource and waiting to acquire additional resources held by the other process.

(3) No preemption:

* A process is holding a resource, then that resource cannot be taken away from that process until the process voluntarily release it.

(4) circular wait:

* A set of processes ($P_0, P_1, P_2 \ldots P_n$) must exist such that P_0 is waiting for a resource that is held by P_1
P_0 is waiting for a resource that is held by P_2.

(2) A data base is to be shared among several concurrent process.

* Readers - processes that only read the database.
* writers - processes perforning both read and write.
* problem: If two readers access the shared data simultaneously no problem will result. But if a writer and some other thread access the database simultaneously, problem arises.
* This synchronization problem is referred to as the readers writers problem.
* The readers - writers problem has several variations.
* The first readers : writers problem, which requires that no reader must be kept waiting unless a writer has already obtained permission to use the shared object.
* The second readers-writers problem requires that once a writer is ready, that writer perform its write as soon as possible.
* The semaphore wrt is common to both readers and writers processes.
* The mutex semaphore is used to ensure mutual exclusion when the variable read count is updated.
* The read count variable keeps track of how many processes are currently reading the object.
* The semphore wrt functions as a mutual exclusion semaphore for the writers.

* The structure of writer process

```
do {
    wait (rw_mutex);
    ------
    /* writing is performed */
    ----
    Signal (rw_mutex);
} while (true);
```

* The structure of reader process

```
do {
    wait (mutex);
    read_count ++ ;
    if (read_count == 1)
    wait (rw_mutex);
    Signal (mutex);
    /* reading is performed */
    ------
    wait (mutex);
    read_count --;
    if (read_count == 0)
        Signal (rw_mutex);
    Signal (mutex);
} while (true)
```

Unit Test - 04

OS (18CS43)

Shahul Hameed s
1KN18CS097
CSE 'A' Sec
4th Sem.

(1) For the following page reference string 1,2,3,4,1,2,5,1,2,3,4,5 Calculate the page faults using FIFO, LRU for memory with 3 and 4 frames.

FIFO Example (3 frames)

Reference String : 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5

access 1 - (1) fault
access 2 - (1,2) fault
access 3 - (1,2,3) fault
access 4 - (2,3,4) fault, replacement
access 1 - (3,4,1) fault, replacement
access 2 - (4,1,2) fault, replacement
access 5 - (1,2,5) fault, replacement
access 1 - (1,2,5)
access 2 - (1,2,5)
access 3 - (2,5,3) fault, replacement
access 4 - (5,3,4) fault, replacement
access 5 - (5,3,4)

- 9 page faults,

LRU Example (3 frames)

- reference string : 1,2,3,4,1,2,5,1,2,3,4,5

access 1 - (1) fault
access 2 - (1,2) fault
access 3 - (1, 2, 3) fault
access 4 - (2, 3, 4) fault, replacement.
access 1 - (3, 4, 1) fault, replace ment
access 2 - (4, 1, 2) fault, replacement
access 5 - (1, 2, 5) fault, replacement
access 1 - (1, 2, 5)
access 2 - (1, 2, 5)
access 3 - (2, 5, 3) fault replacement
access 4 - (5, 3, 4) fault replacement
access 5 - (5, 3, 4)

- 10 page faults.

FIFO Example (4 frames)

- reference string : 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5
access 1 - (1) fault
access 2 - (1, 2) fault
access 3 - (1, 2, 3) fault
access 4 - (1, 2, 3, 4) fault, replacement
access 1 - (2, 3, 4, 1)
access 2 - (3, 4, 1, 2)
access 5 - (4, 1, 2, 5) fault, replacement
access 1 - (4, 2, 5, 1)
access 2 - (4, 5, 1, 2)
access 3 - (5, 1, 2, 3) fault, replacement
access 4 - (1, 2, 3, 4) fault, replacement
access 5 - (2, 3, 4, 5) fault, replacement.

- 8 faults.

LRU Example (4 frames)

- Reference String : 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5

access 1 - (1) fault
access 2 - (1, 2) fault
access 3 - (1, 2, 3) fault
access 4 - (1, 2, 3, 4) fault, replacement
access 1 - (1, 2, 3, 4)
access 2 - (1, 2, 3, 4)
access 5 - (2, 3, 4, 5) fault, replacement
access 1 - (3, 4, 5, 1) fault, replacement
access 2 - (4, 5, 1, 2) fault, replacement
access 3 - (5, 1, 2, 3) fault, replacement
access 4 - (1, 2, 3, 4) fault, replacement
access 5 - (2, 3, 4, 5) fault, replacement.

10 page faults.

(6) Illustrate how demand paging affects System performance

If the number of page faults is equal to the number of referred pages or the no of pg faults are so high. So that the cpu remains busy in just reading the pages from Secondary memory then the effective access time will be taken by the cpu to read one word from the Secondary memory and it will be so high. This concept is called Thrashing.

If the pg fault rate is p f.r., the time taken in getting a page from the Secondary memory and again restarting is s (service time) and the memory access time is (ma) then the effective access time can be given as

$$EAT = PF \times s + (1 - PF) \times (ma)$$

Unit test - 05

OS (18CS43)

Shahul Hameed. S
1KN18CS097
CSE 'A' Sec
4th Sem.

(1) Explain bad-block recovery in disk.

> Bad block is an area of storing media that is no longer reliable for the storage of data because it is completely damaged ~~or~~/corrupted.

Causes of bad-block :- storage devices can ship from the factory with defective blocks that ~~developing~~ originated in the manufacturing process. The device with ~~blocks~~ are marked as defective before leaving the factory.

Types of bad blocks are :-

(1) physical bad block / hard block :- ~~it~~ comes from damage to the storage medium.

(2) Soft / logical bad block :- A ~~soft~~, or logical bad block occurs when the OS is unable to read data from a sector.

(2) Explain the way process is managed by LINUX platform.

> A process means program in execution. It generally takes an i/p, process it and gives us the appropriate o/p. There are basically 2 types.

(1) foreground process :- Such kind of processes are also known as interactive processes. These are the processes which are to be executed / initiated by the user

/the programmer. they can't be initialized by System Services. Such processes take I/p from the user and return the O/p. while these processes are running we can't directly initiate a new process from the same terminal.

② Background processes:-

Such kind of processes are also known as non interactive processes. These are the processes that are to be executed or initiated by the system itself or by users, though they can even be managed by users.

These processes will have a unique PID/process id assigned to them and we can initiate other processes within the same terminal from which they are initiated.