☺😐☹ **MUHAMMAD FAISAL** ☺😐☹

MIT 4th Semester

Al-Barq Campus (VGJW01) Gujranwala

faisalgrw123@gmail.com

Reference Short Questions for Final TERM EXAMS

CS604 – OPERATING SYSTEM

Q.No.1 Scan Algorithm is sometimes called the elevator algorithm, why? (Page#244)

Answer:-

The Scan algorithm is sometimes called the elevator algorithm, since the disk arm behaves like an elevator in a building servicing all the requests (people at floors), going up and then reversing to service the requests going down.

Q.No.2 What is basic logic in FIFO page replacement algorithm? (Page#199)

Answer:-

The simplest page-replacement algorithm is a FIFO algorithm. A FIFO replacement algorithm associates with each page the time when that page was brought into memory. When a page must be replaced, the oldest page is chosen.

Q.No.3 What is mounting? Name two types of mounting. Give your answer with respect to File System? (Page#226)

Answer:-

Mounting makes file systems, files, directories, devices, and special files available for use at a particular location. Mount point is the actual location from which the file system is mounted and accessed. You can mount a file or directory if you have access to the file or directory being mounted and write permission for the mount point

These are the types of mounts:
1. Remote mount
2. Local mount

Q.No.4 Write three main characteristics memory management System? (Page#151)

Answer:-

1) The purpose of memory management is to ensure fair, secure, orderly, and efficient use of memory.
2) The task of memory management includes keeping track of used and free memory space, as well as when, where, and how much memory to allocate and de-allocate.
3) It is also responsible for swapping processes in and out of main memory.

Q.No.5 Summarize the tradeoffs among simple arrays, trees, and hash tables as implementations of a page table. (Page#173,186)

Answer:-

Arrays: Arrays, lists and tables are often allocated more memory than they actually need. An array may be declared 100 by 100 elements even though it is seldom larger than 10 by 10 elements.

Hash Tables: This is a common approach to handle address spaces larger then 32 bits .Usually open hashing is used. Each entry in the linked list has three fields: page number, frame number for the page, and pointer to the next element

Q.No.6 How to implement hold and wait which can ensure that a deadlock will not occur? (Page#129)

Answer:-

A process must be holding at least one resource and waiting to acquire additional resources that are currently being held by other processes.

Q.No.7 List down 2 major benefits of virtual memory (Page#186)

Answer:-

1. Virtual Memory is the separation of user logical memory from physical memory. This separation allows an extremely large virtual memory to be provided for programmers when only a smaller physical memory is available.

2. Virtual memory makes the task of programming easier because the programmer need not worry about the amount of physical memory.

Q.No.8 What are the possible system for the input redirection in the UNIX/LINX system (Page#55)

Answer:-

Linux redirection features can be used to detach the default files from stdin, stdout, and stderr and attach other files with them for a single execution of a command. The act of detaching defaults files from stdin, stdout, and stderr and attaching other files with them is known as input, output, and error redirection.

Here is the syntax for input redirection:
command < input-file or command 0< input-file

Q.No.9 What is the purpose of "stub" in dynamic linking, give answer with respect to memory? (Page#155)

Answer:-

With dynamic linking, a stub is included in the image for each library-routine reference. This stub is a small piece of code that indicates how to locate the appropriate memory-resident library routine or how to load the library if the routine is not already present. During execution of a process, stub is replaced by the address of the relevant library code and the code is executed .If library code is not in memory, it is loaded at this time.

Q.No.10 Dynamic linking (Page#156)

Answer:-

Dynamic linking requires potentially less time to load a program. Less disk space is needed to store binaries. However it is a time-consuming run-time activity, resulting in slower program execution.

Q.No.11 What is use of mounting in file system? (Page#226)

Answer:-

Mounting makes file systems, files, directories, devices, and special files available for use at a particular location. Mount point is the actual location from which the file system is mounted and accessed.

Q.No.12 How operating attacks the" no preemption "condition necessary for feedback in order to solve the pro of deadlock? (Page#129)

Answer:-

No preemption:
Resources cannot be preempted. That is, after using it a process releases a resource only voluntarily.

Q.No.13 What is pager? Give answer with respect to virtual memory (Page#187)

Answer:-

A pager is concerned with the individual pages of a process. Thus the term pager is used in connection with demand paging.

When a process is to be swapped in, the paging software guesses which pages would be used before the process is swapped out again. Instead of swapping in a whole process, the pager brings only those necessary pages into memory.

Q.No.14 What does the following command do in the LINX/UNIX operating system. (Page#26)

Answer:-

$mkdir ~/courses/cs604/program
Command creates the programs directory under your ~/courses/cs604 directory.

Q.No.15 How you can differentiate between external and internal fragmentation

Answer:-

Fragmentation occurs in a dynamic memory allocation system when many of the free blocks are too small to satisfy any request.

External Fragmentation: External Fragmentation happens when a dynamic memory allocation algorithm allocates some memory and a small piece is left over that cannot be effectively used. If too much external fragmentation occurs, the amount of usable memory is drastically reduced. Total memory space exists to satisfy a request, but it is not contiguous.

Internal Fragmentation: Internal fragmentation is the space wasted inside of allocated memory blocks because of restriction on the allowed sizes of allocated blocks. Allocated memory may be slightly larger than requested memory; this size difference is memory internal to a partition, but not being used

Reference: http://wiki.answers.com/Q/What_is_the_difference_between_external_and_internal_fragmentation

Q.No.16 How page fault frequency can b used as a method of thrashing. (Page#211)

Answer:-

Page fault frequency is another method to control thrashing. Since thrashing has a high page fault rate, we want to control the page fault frequency. When it is too high we know that the process needs more frames. Similarly if the page-fault rate is too low, then the process may have too many frames. The operating system keeps track of the upper and lower bounds on the page-fault rates of processes. If the page-fault rate falls below the lower limit, the process loses frames. If page-fault rate goes above the upper limit, process gains frames. Thus we directly measure and control the page fault rate to prevent thrashing.

Q.No.17 Three major frame allocation schemes? (Page#205)

Answer:-

There are three major allocation schemes:

1. **Fixed allocation:** In this scheme free frames are equally divided among processes.

2. **Proportional Allocation:** Number of frames allocated to a process is proportional to its size in this scheme.

3. **Priority allocation:** Priority-based proportional allocation

Q.No.18 Consider the round robin technique .do u think that the deadlock or starvation can happen in the round robin tech scheduling (Page#86)

Answer:-

No I don't think so that the deadlock or starvation can happen in the round robin tech scheduling, because round-robin (RR) scheduling algorithm is designed especially for time-sharing systems. It is similar to FCFS scheduling but preemption is added to switch between processes. A small unit of time, called a time quantum (or time slice) is defined. The ready queue is treated as a circular queue. The CPU scheduler goes around the ready queue, allocating the CPU to each process for a time interval of up to 1 time quantum.

Q.No.19 Explain the work of copy on write with respect to virtual memory. (Page#194)

Answer:-

Many child processes invoke the exec () system call immediately after creation, the copying of the parent's address space may be unnecessary. Alternatively we can use a technique known as copy on write. This works by allowing the parent and child processes to initially share the same pages. These shared pages are marked as copy-on-write pages, meaning that if either process writes to a shared page, a copy of the shared page is created.

Q.No.20 Context switching (Page#31)

Answer:-

Switching the CPU from one process to another requires saving of the context of the current process and loading the state of the new process, this is called context switching.

Q.No.21 Basic logic in FIFO page replacement algorithm (Page#199)

Answer:-

The simplest page-replacement algorithm is a FIFO algorithm. A FIFO replacement algorithm associates with each page the time when that page was brought into memory. When a page must be replaced, the oldest page is chosen. Notice that it is not strictly necessary to record the time when a page is brought in. We can create a FIFO queue to hold all pages in memory. We replace the page at the head of the queue. When a page is brought into memory we insert t at the tail of the queue.

Q.No.22 Formula to find size of page table, (Page#166)

Answer:-

Page table size = NP * PTES
Where NP is the number of pages in the process address space and PTES is the page table entry size

Q.No.23 File control block (Page#233)

Answer:-

A file control block is a memory data structure that contains most of the attributes of a file. In UNIX, this data structure is called inode (for index node).

Q.No.24 One of the responsibilities of O.S is to use computer hardware efficiently, so look Algorithm for disk scheduling, (Page#243)

Answer:-

One of the responsibilities of the operating system is to use the computer system hardware efficiently. For the disk drives, meeting this responsibility entails having a fast access time and disk bandwidth.
I. Structure of 2-level page table,
ii. If a process exits but its threads are still running, will they continue?

Q.No.25 One advantage and one disadvantage of using a large block size to store file data (NET),

Answer:-

Advantage:

- Has lower overhead, so there is more room to store data.
- Good for sequential access or very large rows
- Permits reading a number of rows into the buffer cache with a single I/O (depending on row size and block size).

Disadvantage:

- Wastes space in the buffer cache, if you are doing random access to small rows
- And have a large block size.
- Not good for index blocks used in an OLTP

Q.No.26 Three types of access modes and classes of users in UNIX protection, (Page#230)

Answer:-

UNIX recognizes three modes of access: read, write, and execute (r, w, x). The execute permission on a directory specifies permission to search the directory.

The three classes of users are:

- Owner: user is the owner of the file
- Group: someone who belongs to the same group as the owner
- Others: everyone else who has an account on the system

Q.No.27 Possible criteria to decide that which process should be terminated while dead lock detection and recovery. (Page#149)

Answer:-

When a deadlock detection algorithm determines that a deadlock exists, several alternatives exist. One possibility is to inform the operator that a deadlock has occurred, and to let the operator deal with the deadlock manually. The other possibility is to let the system recover from the deadlock automatically. There are two options for breaking a deadlock. One solution is simply to abort one or more processes to break the circular wait. The second option is to preempt some resources from one or more of the deadlocked processes

Q.No.28 What is mounting in the file system? And where is the mount point? What is mounting? And what is Mount Point? (Page#226)

Answer:-

Mounting makes file systems, files, directories, devices, and special files available for use at a particular location. Mount point is the actual location from which the file system is mounted and accessed. You can mount a file or directory if you have access to the file or directory being mounted and write permission for the mount point.

Q.No.29 Define Roll in & Roll out with respect to swapping (Page#159)

Answer:-

A process needs to be in the memory to be executed. A process, however, can be swapped temporarily out of memory to a backing store, and then brought back into memory for continued execution. Backing store is a fast disk large enough to accommodate copies of all memory images for all users; it must provide direct access to these memory images. The system maintains a ready queue of all processes whose memory images are on the backing store or in memory and are ready to run. This technique is called roll out, roll in.

Q.No.30 Explain the FIFO page algorithm with a scenario where the Belady's anomaly true (Page#199)

Answer:-

The problem with this algorithm is that it suffers from Belady's anomaly: For some page replacement algorithms the page fault rate may increase as the number of allocated 199 frames increases, whereas we would expect that giving more memory to a process would improve its performance.

Q.No.31 Differentiate between dead lock avoidance and dead lock prevention (Page#133)

Answer:-

Deadlock Prevention	Deadlock Avoidance:
➢ Preventing deadlocks by constraining how requests for resources can be made in the system and how they are handled (system design). ➢ The goal is to ensure that at least one of the necessary conditions for deadlock can never hold.	➢ The system dynamically considers every request and decides whether it is safe to grant it at this point. ➢ The system requires additional apriority information regarding the overall potential use of each resource for each process. ➢ Allows more concurrency.

Reference: http://www.cs.jhu.edu/~yairamir/cs418/os4/tsld011.htm

Q.No.32 Write one advantage and one disadvantage of using large size block

Answer:-

Advantages:-
If you use larger block then relatively less overhead. Per I/O you can fetch more data. This is very good for sequential access, or very large rows.

Disadvantages:-
Large block size is not good for index blocks used in an OLTP(Online Transaction Processing) type environment, because they increase block contention on the index leaf blocks.

Reference: http://arjudba.blogspot.com/2008/06/advantages-and-disadvantages-of-using.html

Q.No.33 Does cooperating processes are helpful to the operating system? (Page#41)

Answer:-
The concurrent processes executing in the operating system may be either independent processes or cooperating processes. A process is independent if it cannot affect or be affected by any other process executing in the system. Clearly any process that shares data with other processes is a cooperating process.

Q.No.34 How page fault frequency model used to control the thrashing? (Page#211)

Answer:-

Page fault frequency is another method to control thrashing. Since thrashing has a high
page fault rate, we want to control the page fault frequency. When it is too high we know that the process needs more frames. Similarly if the page-fault rate is too low, then the process may have too many frames.

Establish "acceptable" page-fault rate

- If rate too low, process loses frame
- If rate too high, process gains frame

Q.No.35 What is swap space? (Page#34)

Answer:-

The area on the disk where swapped out processes are stored is called the swap space.

Q.No.36 Writes the formula to calculate page table size (Page#166)

Answer:-

Page table size = NP * PTES , where NP is the number of pages in the process address space and PTES is the page table entry size (equal to |f| based on our discussion so far).

Q.No.37 Name of two registers used in segmentation (Page#176)

Answer:-

Segment-table base register (STBR)
Segment-table length register (STLR) indicates

Q.No.38 What is lazy analyzer that is use in virtual memory? (Page#187)

Answer:-

When we want to execute a process, we swap it into memory. Rather than swapping the entire process into memory, however we use a lazy swapper

Q.No.39 How to detect cycles in acyclic graph? (Page#224)

Answer:-

A solution is to allow only links to files not subdirectories. Also every time a new link is added use a cycle detection algorithm to determine whether it is OK. If cycles are allowed, we want to avoid searching any component twice. A similar problem exists when we are trying to determine when a file can be deleted.

Q.No.40 Name three access modes use in file protection (Page#178)

Answer:-

1. Read
2. Write
3. Execute

Q.No.41 How to detect and recover deadlock? (Page#133)

Answer:-

One method is to allow the system to enter a deadlocked state, detect it, and recover.

Q.No.42 How semaphore algorithm works in n-critical problem? (Page#109,110)

Answer:-

We can use semaphores to deal with the n-process critical section problem. The n processes share a semaphore, **mutex** (standing for mutual exclusion) initialized to 1. Each process Pi is organized as follows:

```
do
{
wait(mutex);
Critical section
signal(mutex);
Remainder section
} while(1);
```

As was the case with the hardware-based solutions, this is not a good solution because even though it satisfies mutual exclusion and progress, it does not satisfy bounded wait. In a uni-processor environment, to ensure atomic execution, while executing wait and signal, interrupts can be disabled

Q.No.43 Some way to reduce external fragmentation (Page#235)

Answer:-

External fragmentation of disk (similar to external fragmentation of main memory due to segmentation). Disk defragmenter utility needs to be used for removing external fragmentation.

Q.No.44 Four characteristic of deadlock prevention (Page#133,134)

Answer:-

1) Mutual exclusion
2) Hold and Wait
3) No preemption
4) Circular Wait

Q.No.45 is starvation and deadlock are same. Accept or reject with solid reason? (Page#113,120)

Answer:-

No! Two neighbors are eating simultaneously; it nevertheless must be rejected because it has the possibility of creating a deadlock.

A set of processes are said to be in a deadlock state if every process is waiting for an event that can be caused only by another process in the set and Starvation is infinite blocking caused due to unavailability of resources.

Q.No.46 Why we need medium term scheduling? (Page#34)

Answer:-

Medium-term scheduler, which removes processes from memory (and from active contention for the CPU) and thus reduces the degree of multiprogramming.

Q.No.47 Difference between deadlock avoidance and deadlock (snowz) prevention? (Page#133)

Answer:-

Deadlock prevention: is a set of methods for ensuring that at least one of the necessary conditions cannot hold. These methods prevent deadlocks by constraining how processes can request for resources.

Deadlock Avoidance: This method of handling deadlocks requires that processes give advance additional information concerning which resources they will request and use during their lifetimes. With this information, it may be decided whether a process should wait or not.

Q.No.48 What is lazy analyzer that is use in virtual memory? (Page#187)

Answer:-

When we want to execute a process, we swap it into memory. Rather than swapping the entire process into memory, however we use a lazy swapper.

Q.No.49 List down 2 major benefits of virtual memory (Page#186)

Answer:-

1. Virtual Memory is the separation of user logical memory from physical memory. This separation allows an extremely large virtual memory to be provided for programmers when only a smaller physical memory is available.

2. Virtual memory makes the task of programming easier because the programmer need not worry about the amount of physical memory

Q.No.50 Possible criteria to decide that which process should be terminated while dead lock detection and recovery? (Page#149)

Answer:-

When a deadlock detection algorithm determines that a deadlock exists, several alternatives exist. One possibility is to inform the operator that a deadlock has occurred, and to let the operator deal with the deadlock manually. The other possibility is to let the system recover from the deadlock automatically. There are **two options** for breaking a deadlock.

1. One solution is simply to abort one or more processes to break the circular wait.
2. The second option is to preempt some resources from one or more of the deadlocked processes

Q.No.51 Some times mkfifo call may be failure,write the reasons for failure if mkfifo call in Unix? (Page#57)

Answer:-

Some of the reasons for this call to fail are:

1) File with the given name exists
2) Pathname too long
3) A component in the pathname not searchable, non-existent, or non-directory
4) Destination directory is read-only
5) Not enough memory space available
6) Signal caught during the execution of mknod()

Q.No.52 What factors are determine to choose a process for Termination? (Page#148,149)

Answer:-

Many factors determine which process is chosen, including:

1) What the priority of the process is
2) How long the process has computed, and how much longer the process will compute before completing its designated task.
3) How many and what type of resources the process has used
4) How many resources the process needs in order to complete
5) How many processes will need to be terminated
6) Whether the process is interactive or batch

Q.No.53 An address that is generated by CPU what is called? (Page#153)

Answer:-
An address generated by the CPU is commonly referred to as a logical address.

Q.No.54 What steps are needed for page replacement? (Page#197)

Answer:-

Steps needed for page Replacement is

1. Find the location of the desire page on the disk
2. Find a free frame
    a. If there is a free frame use it.
    b. If there is no free frame ,use a page replacement algorithm to select a victim frame
3. read the desired page into the newly freed frame; change the page and frame tables
4. Restart the user Process.

QNo.55 Define mounting in UNIX) (Page#226)

Answer:-
Mounting makes file systems, files, directories, devices, and special files available for use at a particular location.

Q.No.56 CPU send the user or application request to the kernel. How kernels work on the request and fulfill it? (Page#14)

Answer:-
There are four events that cause execution of a piece of code in the kernel.

These events are:

- **a.** Interrupt
- **b.** Trap
- **c.** System call
- **d.** Signal.

Some kernel code is executed to service the corresponding event

Q.No.57 Write the steps for converting source code to executable form. (Page#152)

Answer:-
Translation of a source program in a high-level or assembly language involves compilation and linking of the program series of steps:

Compile/Assemble
↓
Link
↓
Load
↓
Execute

Q.No.58 UNIX 3 modes of access. (Owner, group, others) (Page#230)

Answer:-
UNIX recognizes three modes of access: read, write, and execute (r, w, x). The execute permission on a directory specifies permission to search the directory.
The three classes of users are:

Owner: user is the owner of the file
Group: someone who belongs to the same group as the owner
Others: everyone else who has an account on the system

Q.No.59 Fg,bg commands (Page#66)

Answer:-

$ bg
[1]+ find / -name foo -print 2> /dev/null &
$ fg
find / -name foo -print 2> /dev/null

Q.No.60 Disk scheduling SCAN and Look (Page#247)

Answer:-

In the C-Scan and C-Look algorithms, when the disk head reverses its direction, it moves all the way to the other end, without serving any requests, and then reverses again and starts serving requests. In other words, these algorithms serve requests in only one direction.

Q.No.61 Hierarchical paging in Intel 80386 (Page#171)

Answer:-

Most modern computers support a large logical address space: (232 to 264). In such an environment, the page table itself becomes excessively large

QNo.62 Load time and compile time linking, execution time linking (dynamic and static Linking) (Page#152,155)

Answer:-

Compile time: if you know at compile where the process will reside in memory, the absolute addresses can be assigned to instructions and data by the compiler.

Load time: if it is not known at compile time where the process will reside in memory, then the compiler must generate re-locatable code. In this case the final binding is delayed until load time.

Execution time: if the process can be moved during its execution from one memory segment to another, then binding must be delayed until run time. Special hardware must be available for this to work. The size of a process is limited to the

size of physical memory. To obtain better memory space utilization, we can use dynamic loading.

Q.No.63 Safe and unsafe state, safe sequence (Page#135)

Answer:-

A state is safe if the system can allocate resources to each process in some order and still avoid a deadlock. More formally a system is in a safe state only if there exists a safe sequence. A sequence of processes <P1, P2… Pn> is a safe sequence for the current allocation state if, for each Pi, the resources that Pi can still request can be satisfied by the currently available resources plus all the resources held by all the Pj with j < i. In this situation, if the resources that Pi needs are not immediately available, then Pi can wait until all Pj have finished. When they have finished, Pi can obtain all of its needed resources, complete its designated task, return its allocated resources and terminate. When Pi terminates, Pi+1 can obtain its needed resources and terminate. If no such sequence exists, then the system is said to be unsafe.

Q.No.64 Hold and wait condition for Deadlocks (Page#129)

Answer:-

A process must be holding at least one resource and waiting to acquire additional resources that are currently being held by other processes.

Q.No.65 Difference between Bounded wait and progress condition. (Page#99)

Answer:-

Process:- If no process is executing in its critical section and some processes wish to enter their critical sections, then only those processes that are not executing in their remainder section can participate in the decision on which will enter its critical section next, and this selection cannot be postponed indefinitely.

Bounded Waiting:- There exists a bound on the number of times that other processes are allowed to enter their critical sections after a process has made a request to enter its critical section and before that request is granted.

Q.No.66 Define logical address (Page#153)

Answer:-
An address generated by the CPU is commonly referred to as a logical address.

Q.No.67 Real time system definition (Page#06)

Answer:-
Real time systems are used when rigid time requirements are placed on the operation of a processor or the flow of data; thus it is often used as a control device in a dedicated application.

Q.No.68 Issue with segmentation and solution (Page#179)

Answer:-

Segmentation may then cause external fragmentation (i.e. total memory space exists to satisfy a space allocation request for a segment, but memory space is not contiguous), when all blocks of memory are too small to accommodate a segment. In this case, the process may simply have to wait until more memory (or at least a larger hole) becomes available or until compaction creates a larger hole. Since segmentation is by nature a dynamic relocation algorithm, we can compact memory whenever we want. If we define each process to be one segment, this approach reduces to the variable sized partition scheme. T the other extreme, every byte could be put in its own segment and relocated separately. This eliminates external fragmentation altogether, however every byte would need a base register for its relocation, doubling memory use. The next logical step- fixed sized, small segments, is paging i.e. paged segmentation. Also it might latch a job in memory while it is involved in I/O. To prevent this I/O should be done only into OS buffers.

Q.No.69 Problems in round robin with swapping (Page#158)

Answer:-

Swapping is constrained by factors like quantum for RR scheduler and pending I/O for swapped out process. Assume that I/O operation was queued because the device was busy. Then if we were to swap out P1, and swap in process P2, the I/O operation might attempt to access memory that now belongs to P2.The solution to this problem are ever to swap out processes with pending I/O or to execute I/O in kernel space.

Q.No.70 Methods for process communication (Page#44)

Answer:-

A link is established automatically between every pair of processes that want to communicate. The processes need to know only each other's identity to communicate

- A link is associated with exactly two processes.
- Exactly one link exists between each pair of processes

Q.No.71 Five attributes of File (Page#51)

Answer:-

1) File Descriptor
2) Per Process File Descriptor Table
3) File Tablee
4) Inode Table
5) File's contents

Q.No.72 Mmap() command (Page#196)

Answer:-

"Normal" File I/O
fildes = open(...);
lseek(...);
read(fildes, buf, len);
/* use data in buf */
File I/O with mmap()
fildes = open(...)
address = mmap((caddr_t) 0, len,(PROT_READ | PROT_WRITE), MAP_PRIVATE, fildes, offset);
/* use data at address */

Q.No.73 The maximum number of pages in process address space is one million and the total address size (p +d) of process address space is 32- bit with page size is 4096 bytes. Calculate the number of bits required for page number (p) and the number of bits required for offset (d)?

Answer:-

P = 12 bits
Off set = 5
Let the number of bits required = x
so, $2^x = 4096$
so, $x = 12$
$Offset = 5$
$2^p = 4096$
$2^p = 2^{12}$
$p = 12$
$2^d = 32$
$2^d = 2^5$
$d = 5$

QNo.74 What are the three stages/times when the address is bound to instructions and data? (Page#152)

Answer:-

Address can be bound to instructions and data at different times, as discussed below briefly.

1. **Compile time:** if you know at compile where the process will reside in memory, the absolute addresses can be assigned to instructions and data by the compiler.

2. **Load time:** if it is not known at compile time where the process will reside in memory, then the compiler must generate re-locatable code. In this case the final binding is delayed until load time.

3. **Execution time:** if the process can be moved during its execution from one memory segment to another, then binding must be delayed until run time. Special hardware must be available for this to work.

Q.No.75 What is mounting and explain It.? (Page#226)

Answer:-

Mounting makes file systems, files, directories, devices, and special files available for use at a particular location. Mount point is the actual location from which the file system is mounted and accessed. You can mount a file or directory if you have access to the file or directory being mounted and write permission for the mount point

There are types of mounts:

1. Remote mount
2. Local mount

QNo.76 What is Load Time? (Page#226)

Answer:-

If it is not known at compile time where the process will reside in memory, then the compiler must generate re-locatable code. In this case the final binding is delayed until load time.

QNo.77 What is Index Allocation Method? (Page#236)

Answer:-

Indexed allocation brings all the pointers to the block together into a disk block, known as the index block.

QNo.78 Choose Preemptive and Non-Preemptive Scheduling from the following;
First-Come-First-Served (FCFS) Scheduling
Shorted Job First (SJF) Scheduling
Shortest Remaining Time First (SRTF) Scheduling
Priority Scheduling
Round-Robin Scheduling
Multilevel Queues Scheduling
Multilevel Feedback Queues Scheduling

Answer:-

Preemptive and	Non-Preemptive
➢ Shortest Remaining Time First (SRTF) Scheduling ➢ Round-Robin Scheduling	➢ First-Come, First Serve (FCFS or FIFO ➢ Shorted Job First (SJF) Scheduling

Reference: http://academic.udayton.edu/SaverioPerugini/courses/cps346/lecture_notes/scheduling.html

QNo.79 How can u display the status of suspended and background processes in Unix/Linux shell? (Page#65)

Answer:-

You can use the fg command to resume the execution of a suspended job in the foreground or move a background job into the foreground. Here is the syntax of the command.
fg [%job_id]

Q.No.80 Consider a logical address space of eight pages of 1024 words each, mamapped ont a Physical memory of 32 frames.
(a)How many bits are there in the logical address?
(b)How many bits are there in the physical address?

Answer:-

Each page/frame holds 1K; we will need 10 bits to uniquely address each of those 1024 addresses. Physical memory has 32 frames and we need 2 5 bits to address each frames, requiring in total 5+10=15 bits. A logical address space of 64 pages requires 6 bits to address each page uniquely, requiring 16bits in total.

Reference: http://garryowen.csisdmz.ul.ie/~cs4023/resources/sol11.pdf

Q.No.81 Drawbacks of semaphore

Answer:-

- Simple algorithms require more than one semaphore
- This increases the complexity of semaphore solutions to such algorithm
- The programmer must keep track of all calls to wait and to signal the semaphore.
- Since semaphores can be tricky, can we create other constructs from them?

Reference: http://www.cs.colostate.edu/~cs551/CourseNotes/ConcurrentConstructs/DisAdvSems.html

Q.No.82 Address Generated by CPU (Page#153)

Answer:-

An address generated by the CPU is commonly referred to as a logical address.

QNo.83 What is pager with respect to virtual memory? (Page#187)

Answer:-

Whereas a pager is concerned with the individual pages of a process. Thus the term pager is used in connection with demand paging.

Q.No.84 Differ between physical address and virtual address? (Page#153)

Answer:-

Real memory uses Physical addresses. These are the members that the memory chips react to on the bus. Virtual addresses are the logical addresses that refer to a process' address space. Thus, a machine with a 16-bit word can generate virtual addresses up to 64K, regardless of whether the machine has more or less memory than 64 KB.

Q.No.85 Define and briefly describe what is memory mapping system calls? 3calls (Page#196)

Answer:-

The memory mapping system calls can only support copy-on-write functionality allowing processes to share a file in read-only mode, but to have their own copies of any data they modify. So that access to the shared data is coordinated, the processes involved might use one of the mechanisms for achieving mutual exclusion.

mmap() System Call

In a UNIX system, mmap() system call can be used to request the operating system to memory map an opened file. The following code snippets show "normal" way of doing file I/O and file I/O with memory mapped files.

Q.No.86 Consider a process having its segment 15 having 5096 bytes. The process generates a logical address (15, 3921). What page does the logical address refers to? (Page#181)

Answer:-

How many pages does the segment have? ceiling[5096/1024]= 5
What page does the logical address refers to? ceiling[3921/1024]= 4 (i.e., page number 3)

Q.No.87 Disadvantage of dynamic loading? (Page#154)

Answer:-

Disadvantage:-

- However the run time activity involved in dynamic loading is a disadvantage.
- Dynamic programming does not require special support from the operating system.

Q.No.88 Look algorithm? (Page#246)

Answer:-

This algorithm is a version of SCAN. In this algorithm the arm only goes as far as the last request in each direction, then reverses direction immediately, serving requests while going in the other direction. That is, it looks for a request before continuing to move in a given direction.

Q.No.89 Degree of multiprogramming when increased what effect is on CPU utilization? (Page#05)

Answer:-

Multi-programming increases CPU utilization by organizing jobs so that the CPU always has one to execute. The operating system keeps several jobs in memory simultaneously.

Q.No.90 Source open software are helpful for testing algorithms as compared to pirated software?

Answer:-

Open source software is very often developed in a public, collaborative manner. Open source software is the most prominent example of open source development and often compared to (technically defined) user-generated content or (legally defined) open content movements

Reference: http://en.wikipedia.org/wiki/Open_source_software

Q.No.91 Hardware required for swapping, paging and demand paging? (Page#158, 162,186)

Answer:-

Swapping: A process needs to be in the memory to be executed. A process, however, can be swapped temporarily out of memory to a backing store, and then brought back into memory for continued execution. Backing store is a fast disk large enough to accommodate copies of all memory images for all users; it must provide direct access to these memory images. The system maintains a ready queue of all processes whose memory images are on the backing store or in memory and are ready to run.
Paging: Paging is a memory management scheme that permits the physical address space of a process to be non-contiguous. It avoids the considerable problem of fitting the various sized memory chunks onto the backing store, from which most of the previous memory-management schemes suffered. When some code fragments or data residing in main memory need to be swapped out, space must be found on the backing store. The fragmentation problems discussed in connection with main memory are also prevalent with backing store, except that access is much slower so compaction is impossible.

Demand Paging: A demand paging system is similar to a paging system with swapping. Processes reside on secondary memory (which is usually a disk).When we want to execute a process, we swap it into memory. Rather than swapping the entire process into memory, however we use a lazy swapper. A lazy swapper never swaps a page into memory unless that page will be needed. Since we are now viewing a process as a sequence of pages rather than as one large contiguous address space, use of swap is technically incorrect. A swapper manipulates entire processes, whereas a pager is concerned with the individual pages of a process. Thus the term pager is used in connection with demand paging.

QNo.92 Semaphores Algorithm? (Page#107,108)

Answer:-

In this algorithm, we combine the ideas of the first two algorithms. The common data structures used by a cooperating process are:
boolean waiting[n];
boolean lock;
The structure of process Piis:

```
do
{
waiting[i] = true;
key = true;
while (waiting[i] && key)
key = TestAndSet(lock);
waiting[i] = false;
Critical section
j = (i+1) % n;
while ((j!=i) && !waiting[j])
j = (j+1)% n;
if (j == i)
lock = false;
else
waiting[j] = false;
Remainder section
} while(1);
```

These data structures are initialized to false. To prove that the mutual exclusion requirement is met, we note that process Pican enter its critical section only if either waiting[i]= = false or key = = false. The value of key can become false only if TestAndSet is executed. The first process to execute the TestAndSet instruction

will find key= =false; all others must wait. The variable waiting[i] can only become false if another process leaves its critical section; only one waiting[i] is set to false, maintaining the mutual exclusion requirement.
To prove the progress requirement is met, we note that the arguments presented for mutual exclusion also apply here, since a process exiting the critical section either sets lock to false or sets waiting[j] to false. Both allow a process that is waiting to enter its critical section to proceed.

To prove that the bounded waiting requirement is met, we note that, when a process leaves its critical section, it scans the array waiting in the cyclic ordering (i+1, i+2, …, n-1, 0, 1, …, i-1). It designates the first process it sees that is in its entry section with waiting[j]=true as the next one to enter its critical section. Any process waiting to do so will enter its critical section within n-1 turns.

QNo.93 Who generate physical address and logical address? (Page#153)

Answer:-

An address generated by the CPU is commonly referred to as a logical address, where as an address seen by the memory unit–that is, the one loaded into the memory-address register of the memory–is commonly referred to as the physical address. In essence, logical data refers to an instruction or data in the process address space where as the physical address refers to a main memory location where instruction or data resides. The compile time and load time binding methods generate identical logical and physical addresses, where as the execution time binding method results in different physical and logical addresses. In this case we refer to the logical address as the virtual address. The set of all logical addresses generated by a program form the logical address space of a process; the set of all physical addresses corresponding to these logical addresses is a physical address space of the process. The total size of physical
address space in a system is equal to the size of its main memory. The run-time mapping from virtual to physical addresses is done.

Q.No.94 Write the names of common file structures. (Page#215)

Answer:-
A file has certain defined structure characteristics according to its type. A few common types of file structures are:-

1. Simple Record Structures
2. Complex structures

Q.No.95 Which part of compiler/assembler perform the task of taking one or more objects generated and assemble them in to a single executable program. (Page#152)

Answer:-

This process generates the machine language executable code (also known as a binary image) for the give source program. To execute the binary code, it is loaded into the main memory and the CPU state is set appropriately. The whole process is shown in the following diagram.

Compile/Assemble
↓
Link
↓
Load
↓
Execute

QNo.96 Write the method through which Linus/Unix commands can communicate with each other. (Page#26)

Answer:-

Answer $mkdir ~/courses/cs604/program
Command creates the programs directory under your ~/courses/cs604 directory.

Q.No.97 Which term is best suited for the situation where several process access and manipulate shared data concurrently and final value of data depends which process finishes last. (Page#96)

Answer:-

A situation like this, where several processes access and manipulate the same data concurrently and the outcome of the manipulation depends on the particular order in which the access takes place, is called a race condition.

Q.No.98 Possible criteria to decide that which process should be terminated (Page#149)

Answer:-

When a deadlock detection algorithm determines that a deadlock exists, several alternatives exist. One possibility is to inform the operator that a deadlock has occurred, and to let the operator deal with the deadlock manually. The other possibility is to let the system recover from the deadlock automatically. There are two options for breaking a deadlock. One solution is simply to abort one or more processes to break the circular wait. The second option is to preempt some resources from one or more of the deadlocked processes.

Q.No.99 Some times mkfifo call may be failure,write the reasons for failure if mkfifo call in Unix? (Page#57)

Answer:-

Some of the reasons for this call to fail are:

1) File with the given name exists
2) Pathname too long
3) A component in the pathname not searchable, non-existent, or non-directory
4) Destination directory is read-only
5) Not enough memory space available
6) Signal caught during the execution of mknod()

Q.No.100 Which factors are determine to choose a process for Termination? (Page#148,149)

Answer:-

Many factors determine which process is chosen, including:

7) What the priority of the process is
8) How long the process has computed, and how much longer the process will compute before completing its designated task.
9) How many and what type of resources the process has used
10) How many resources the process needs in order to complete
11) How many processes will need to be terminated
12) Whether the process is interactive or batch

Q.No.101 How can you calculate size of page table, give formula? (Page#165)

Answer:-

Page table size = NP * PTES , where NP is the number of pages in the process address space and PTES is the page table entry size (equal to |f| based on our discussion so far).
Page table size = 16 * 5 bits

Q.No.102 Soft links in UNIX? (Page#225)

Answer:-

Soft links take care of all the problems inherent in hard links. They are flexible. You may have soft links to directories and across file systems. However, UNIX has to support an additional file type, the link type, and a new file is created for every link, slowing down file operations.

Q.No.103 A code was given and question was to tell whether it satisfies mutual exclusion?

Answer:-

An algorithm solves the mutual exclusion problem if the following hold:

Mutual Exclusion:
In every configuration of every execution, at most one process is in the critical section.

No Deadlock:
In every execution, if some process is in the entry section in some configuration, then there is a later configuration in which **some process is in the critical section.**
• Stronger Progress Property.

No lockout (starvation-free):
In every execution, if some processor is in the entry section in a configuration, then there is a later configuration in which that same processor is in the critical section.

Reference: http://www.csd.uoc.gr/~hy586/material/lectures/cs586-Section2.pdf

Q.No.104 A diagram was given and question was In fully LRU the concept of aging is used with the algorithm. What does keep track of R-bits at each clock tick?

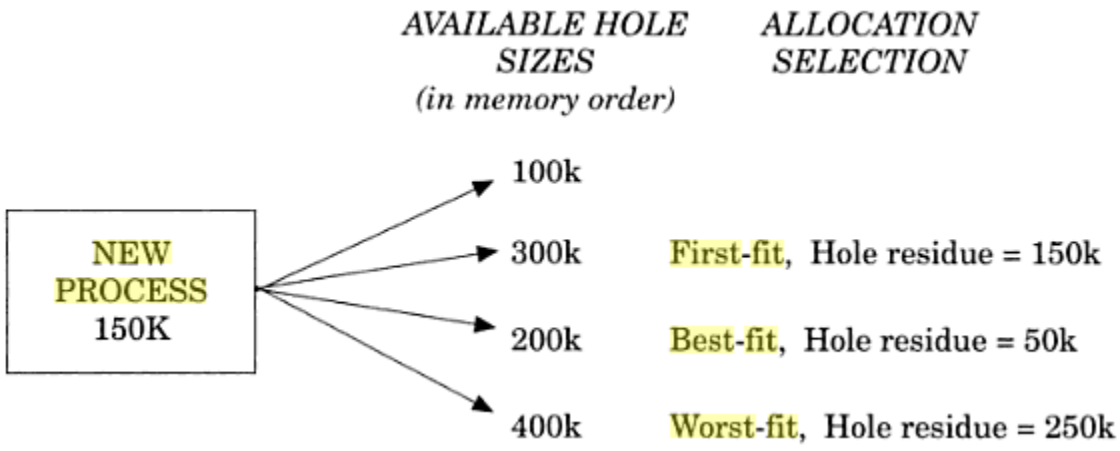
Answer:-

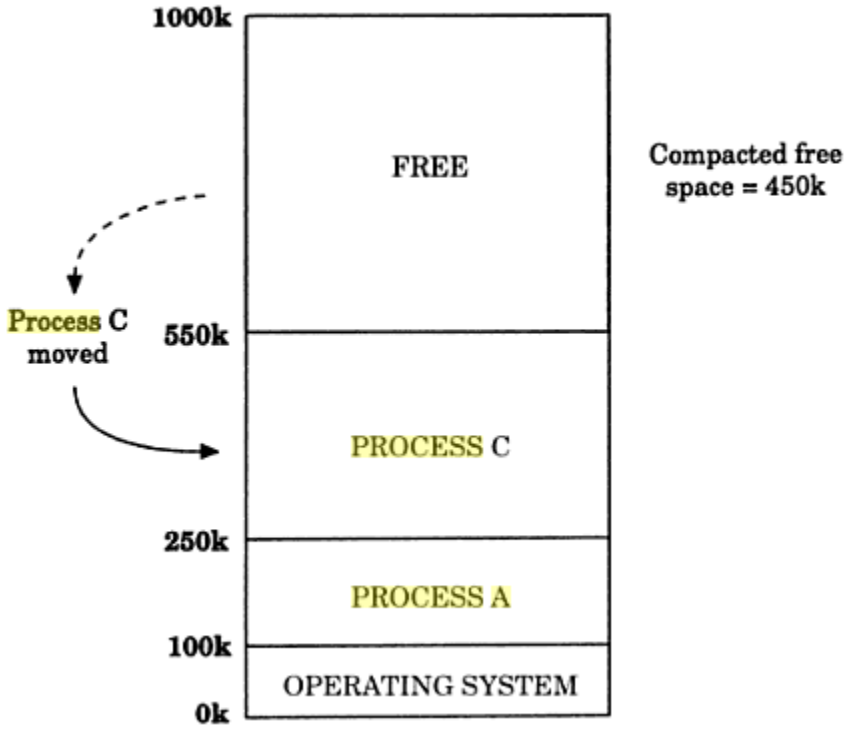
R bits	Page 0	Page 1	Page 2	Page 3
1110	10000000	10000000	10000000	00000000
1001	11000000	01000000	01000000	10000000
1100	11100000	10100000	00100000	01000000
1101	11110000	11010000	00010000	10100000
0010	01111000	01101000	10001000	01010000
1010	10111100	00110100	11000100	00101000
1100	11011110	10011010	01100010	00010100
0011	01101111	01001101	10110001	10001010

Reference: http://www.cs.wichita.edu/~chang/lecture/cs540/homework/hwk3-sol.txt

Q.No.105 If a new process is to be loaded of size 25 k which whole size will be filled using best fit, First fit and worst fit? given hole memory location were given 20k, 15k, 40k, 60k, 10k,25k.

Answer:-





Answer to question 2

Best-fit – 25K, First-fit – 40K, Worst-fit – 60K

Reference:

http://books.google.com.pk/books?id=JmkiE2Ut7ysC&pg=PA85&lpg=PA85&dq=If+a+new+process+is+to+be+loaded+of+size+25+k+which+whole+size+will+be+filled+using+best+fit,+cx`st+fit+and+worst+fit&source=bl&ots=CDtFpjfxg1&sig=TnWpbVD0hxbPV3_dALs7Z8Vp3eU&hl=en&sa=X&ei=fC0OUarKEuqp4gTFoYCQBg&ved=0CC4Q6AEwAA#v=onepage&q=If%20a%20new%20process%20is%20to%20be%20loaded%20of%20size%2025%20k%20which%20whole%20size%20will%20be%20filled%20using%20best%20fit%2C%20First%20fit%20and%20worst%20fit&f=false

Q.No.106 Differentiate between logical and physical address.

Answer:-

Physical Address:

- It is also called as MAC address or Ethernet Address or Layer 2 address.
- This address is burnt in the network Adapter called NIC card.
- This is a 48 bit address represented in hexadecimal format.
- To find known MAC to unknown IP address we need a protocol called RARP-Reverse Address Resolution Protocol.
- This address is used inside LAN networks i.e within a single network.
- Example MAC address: 00-0d-65-ac-50-7f

Logical Address:

- It is also called as Network address or layer 3 address.
- It is a 32 bit address represented in 4 octets.
- This address is used when two or more networks communicate with each other.
- To find unknown IP from MAC address we need a protocol called ARP Address Resolution Protocol.
- This address has two components: Network part and Host part.
- Example IP address: 192.168.1.10

Reference: http://www.indiastudychannel.com/experts/20546-What-difference-between-physical-address.aspx

Q.No.107 Identify the necessary information that must be stored in process control block during the execution of program. (Page#30)

Answer:-

Each process is represented in the operating system by a process control block (PCB) – also called a task control block.

Process state: The state may be new, ready, running, and waiting, halted and so on.

Program counter: The counter indicates the address of the next instruction to be executed for this process.

CPU registers: The registers vary in number and type, depending on the computer architecture. They include accumulators, index registers, stack pointers and general-purpose registers, plus any condition code information. Along with the program counter, this state information must be saved when an interrupt occurs, to allow the process to be continued correctly afterwards.

CPU Scheduling information: This information includes a process priority, pointers to scheduling queues, and any other scheduling parameters.

Memory-management information: This information may include such information such as the value of the base and limit registers, the page tables, or the segment tables, depending on the memory system used by the operating system.

Accounting information: This information includes the amount of CPU and real time used, time limits, account numbers, job or process numbers, and so on.

I/O status information: The information includes the list of I/O devices allocated to the process, a list of open files, and so on.

Best of Luck