



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

Which part of compiler/assembler perform the task of taking one or more objects generated and assemble them in to a single executable program. (2)

Answer:-[Click here for detail](#)

Linking means taking one or more objects generated by compilers and assemble them into a single executable program.

(Page 152)

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

Which term is best suited for the situation where several process access and manipulate shared data concurrently and final value of data depend which process finishes last. (3)

Answer:- (Page 96)

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.

A file system must keep track of free blocks on disk. Name two schemes for doing this, and one advantage for each (5)

Answer:-[Click here for detail](#)

Linked list of free pages supports very fast deletion: we can link a whole file onto the free list in constant time. Bit map facilitates spatial locality optimizations in disk layout. It is also the most efficient in terms of memory inside the OS required to find the free pages.

The list of pages with free page references in each is also space-efficient, particularly for disks that are mostly full (the bit map takes space proportional to the size of the entire disk). Also, free page insertion and deletion operations exhibit high locality under this scheme.

Differentiate between logical and physical address. (5)

Answer:- (Page 152)

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.



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

The compile time and load time binding methods generate identical logical and physical addresses, whereas 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 by a piece of hardware in the CPU, called the memory management unit (MMU).

QNo.2 how to protection of files and giving access to be secured in UNIX ?

Answer:- (Page 229)

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.

QNo.3 what is lazy analyzer that is use in virtual memory?

Or

Define lazy swapper with respect to virtual memory?

Answer:- (Page 186)

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.

QNo.4 Differentiate between Cache and CPU Register?

Answer:-[Click here for detail](#)

CPU register is just a small amount of data storage that facilitates some CPU operations.

CPU cache, it is a high speed volatile memory which is bigger in size, that helps the processor to reduce the memory operations.

QNo.5 what is Index Allocation Method?

Answer:- (Page 235-36)

Indexed allocation brings all the pointers to the block together into a disk block, known as the index block. Indexed allocation supports direct access without suffering from external fragmentation because any free block on the disk may satisfy a request for more space.

QNo.7 What hardware is needed to support demand paging and swapping name and describe briefly?

Answer:- (Page 188)



(Subjective-CS604 OPERATING SYSTEM)

(From Final Term Papers “No repeated Questions”) June 2014

The hardware needed to support demand paging is the same as the hardware for paging and swapping:

- Page table: This table has the ability to mark an entry invalid through a valid-invalid bit or special value of protection bits.
- Secondary memory: This memory holds those pages that are not present in main memory. The secondary memory is usually a high speed disk. It is known as the swap device, and the section of disk used for this purpose is called the swap space.

Why program exhibits that it is always partially in memory (2 marks)

Answer:- (Page 185)

Even in cases where the entire program is needed, it may not be all needed at the same time. The ability to execute a program that is only partially in memory confers many benefits. Running a program that is not entirely in memory would benefit both the system and the user.

Give one advantage and one disadvantage of using large sized block (2)

Answer:-[Click here for detail](#)

Advantage

- Good for large rows with lots of sequential I/O
- Has a lower overhead with less block headers so more room to store data
- Well suited for tables with large sized rows

Disadvantage

- Wastes space in the buffer cache if doing random I/O to access a small number of rows
- In OLTP systems can increase block contention due to more rows being stored in each block

File attributes. Only five (5)

Answer:- (Page 215)

Name: The symbolic file name is the only information kept in human-readable form

Type: This information is needed for those systems that support different types.

Location: This location is a pointer to a device and to the location of the file on that device. Time and date

created: useful for security, protection and usage monitoring.

Time and date last updated: useful for security, protection and usage monitoring.

Whether the files are volatiles or non-volatile? Also comment that whether data can be written on secondary storage if not written in file?

Answer:- (Page 214)

Files usually non-volatile, so the contents are persistent through power failures, etc. A file is a named collection of related information that is recorded on secondary storage. Data cannot be written to secondary storage unless they are within a file.



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

Differentiate between bg and fg command?

Answer:- (Page 65)

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. You can use the bg command to put the current or a suspended process into the background.

Explain how pre-paging prevents high level of initial paging?

Answer:- (Page 211)

An obvious property of a pure demand paging system is the large number of page faults that occur when a process is started. This situation is the result of trying to get the initial locality into memory. Pre-paging is an attempt to prevent this high level of initial paging. The strategy is to bring into memory at one time all the pages that will be needed.

Is it necessary to avoid user program from get stuck in an infinite loop? Give reasons.

Answer:- (Page 11)

In addition to protecting I/O and memory, we must ensure that the operating system maintains control. We must prevent the user program from getting stuck in an infinite loop or not calling system services and never returning control to the CPU. To accomplish this we can use a timer, which interrupts the CPU after specified period to ensure that the operating system maintains control.

The timer period may be variable or fixed. A fixed-rate clock and a counter are used to implement a variable timer. The OS initializes the counter with a positive value. The counter is decremented every clock tick by the clock interrupt service routine. When the counter reaches the value 0, a timer interrupt is generated that transfers control from the current process to the next scheduled process. Thus we can use the timer to prevent a program from running too long.

In the most straight forward case, the timer could be set to interrupt every N milliseconds, where N is the time slice that each process is allowed to execute before the next process gets control of the CPU. The OS is invoked at the end of each time slice to perform various housekeeping tasks.

To resolve thrashing due to excessive paging what can a user do? (2 marks)

OR

How can a user control the Thrashing? Marks (2)

Answer:- (Page 11)

In order to stop thrashing, the degree of multiprogramming needs to be reduced. The effects of thrashing can be reduced by using a local page replacement.

What is the difference between Zero Capacity Buffering and Un-Bounded Capacity Buffering? Marks (2)

Answer:- (Page 44)

Zero Capacity: The queue has maximum length zero, thus the link cannot have any messages waiting in it. In this case the sender must block until the message has been received.



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

Unbounded Capacity: The queue has infinite length; thus the sender never blocks.

What is the basic logic of FIFO Page Replacement Algorithm? Marks (2)

Answer:- (Page 198)

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.

What is the difference between Internal Fragmentation and External Fragmentation Marks (3)?

Answer:-[Click here for detail](#)

Internal Fragmentation is the area in a region or a page that is not used by the job occupying that region or page. This space is unavailable for use by the system until that job is finished and the page or region is released.

What protection built in Primary Protection Scheme under Paging concept? Marks (3)

Answer:- (Page 169)

The primary protection scheme guards against a process trying to access a page that does not belong to its address space. This is achieved by using a valid/invalid (v) bit. This bit indicates whether the page is in the process address space or not. If the bit is set to invalid, it indicates that the page is not in the process's logical address space.

Defined Soft (Symbolic) Link in UNIX. Marks (3)

Answer:- (Page 228)

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.

How can you differentiate a call generated by the system is an Interrupt or Signal? Marks (3)

Answer:- (Page 7)

An interrupt is a signal generated by a hardware device (usually an I/O device) to get CPU's attention.

A trap (or an **exception**) is a software-generated interrupt caused either by an error (division by zero or invalid memory access) or by a user request for an operating system service.

A signal is an event generated to get attention of a process. An example of a signal is the event that is generated when you run a program and then press <Ctrl-C>.

Address translation with Inverted Page Table? Marks (5)

Answer:- (Page 173)



(Subjective-CS604 OPERATING SYSTEM) **(From Final Term Papers “No repeated Questions”) June 2014**

Page table size is limited by the number of frames (i.e., the physical memory) and not process address space. Each entry in the page table contains (pid, p). If a page ‘p’ for a process is loaded in frame ‘f’, its entry is stored at index ‘f’ in the page table. We effectively index the page table with frame number; hence the name inverted page table. Examples of CPUs that support inverted pages tables are 64-bit UltraSPARC and PowerPC.

Briefly describe the concept of the reentrant code?

Answer:- (Page 174)

Reentrant (readonly) code pages of a process address can be shared. If the code is reentrant, it never changes during execution. Thus two or more processes can execute the same code at the same time.

To resolve thrashing due to excessive paging, what can a user do?

Answer:-[Click here for detail](#)

To resolve thrashing due to excessive paging, a user can do any of the following:

- Increase the amount of RAM in the computer (generally the best long-term solution).
- Decrease the number of programs being run on the computer.
- Replace programs that are memory-heavy with equivalents that use less memory.
- Improve spatial locality by replacing loops

How can we calculate a size of page table? Just provide the formula?

Answer:- (Page 165)

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.

Describe the following command Grep "Asif" academics>admin

What are three different stages/times when address can be bound to instructions and data?

Answer:- (Page 151)

Compile time
Load time
Execution time

What is the problem hard links creates in terms of referential naming? How does one solve it?

Answer:-[Click here for detail](#)

When using hard links, we effectively create multiple names for the same data contained within a file. The problem in managing these structures comes down to knowing when the data within the file should actually be



(Subjective-CS604 OPERATING SYSTEM)

(From Final Term Papers “No repeated Questions”) June 2014

deleted. In particular, the data should be deleted only when the last name (or reference) is removed from the file system. This is typically detected through the use of reference counters.

It is also possible to create cycles in the directory structure, which can lead to files and/or directories being left on the disk without any way to get to the from the root directory. This may be solved by either not allowing hard links to directories or by explicitly checking to make sure that a cycle is not created when a link is made to a directory.

If process takes an average page fault service time of 20 milliseconds and a memory access time of 100 nanoseconds, then the effective access time in nanoseconds?

Answer:- (Page 190)

$$\begin{aligned}\text{Effective access time} &= (1-p) * (100) + p (20 \text{ milliseconds}) \\ &= (1-p) * 100 + p * 20,000,000 \\ &= 100 + 19,999,900 * p\end{aligned}$$

If HR is hit ratio and MR is miss ratio, then how can you calculate the effective access time? Write formula and calculate $T_{\text{effective}}$ on following values

$$T_{\text{mem}}=120\text{nsec}$$

$$T_{\text{TTLB}}=20\text{nsec}$$

$$\text{HR}=80\%$$

Answer:- (Page 170)

$$\begin{aligned}T_{\text{effective}} &= \text{HR} (T_{\text{TTLB}} + T_{\text{mem}}) + \text{MR} (T_{\text{TTLB}} + 2T_{\text{mem}}) \\ &= 0.8 (20 + 120) + 0.2 (20 + 2*120) \text{ nanoseconds} \\ &= \mathbf{164 \text{ nanoseconds}}\end{aligned}$$

Analyze the relationship between fork() and except system calls? Let us consider a situation in which we want to communicate unrelated process on the same system?

Answer: - (Page 50)

The BSD sockets are used for communication between related or unrelated processes on the same system or unrelated processes on different systems.

How to detect and recover deadlock?

Answer:- (Page 132)



(Subjective-CS604 OPERATING SYSTEM) **(From Final Term Papers “No repeated Questions”) June 2014**

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

Define Roll in & Roll out with respect to swapping

Answer:- (Page 157)

When the higher--priority process finishes, the lower--priority process can be swapped back in and continued. This technique is called roll out, roll in. The major part of swap time is transfer time; the total transfer time is directly proportional to the amount of memory swapped.

Can we increase the speed by accessing the page table in the memory compare to the CPU register?

Answer:- (Page 166)

A page table base register (PTBR) is needed to point to the page table. With page table in main memory, the effective memory access time, $T_{\text{effective}}$, is $2T_{\text{mem}}$, which is not acceptable because it would slow down program execution by a factor of two.

Explain the FIFO page algorithm with a scenario where the Belady's anomaly true

Answer:- (Page 198)

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 frames increases, whereas we would expect that giving more memory to a process would improve its performance.

If we create a pipe and a pipe return a numeric value -1 what does it means?

Answer:- (Page 46)

The function returns -1 if the call fails.

Does cooperating processes are helpful to the operating system

Answer:- (Page 41)

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.

How page fault frequency model used to control the thrashing

Answer:- (Page 210)

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.



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

What is swap space?

Answer:- (Page 188)

The secondary memory is usually a high speed disk. It is known as the swap device, and the section of disk used for this purpose is called the swap space.

Scan Algorithm is sometimes called the elevator algorithm, why?

Answer:- (Page 244)

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.

What is mounting? Name two types of mounting. Give your answer with respect to File System?

Answer:- (Page 226)

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

Write three main characteristics memory management System?

Answer:- (Page 151)

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 deallocate.
3. It is also responsible for swapping processes in and out of main memory

Summarize the tradeoffs among simple arrays, trees, and hash tables as implementations of a page table.

Answer:- (Page 173,186)

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 than 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



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

How to implement hold and wait which can ensure that a deadlock will not occur?

Answer:- (Page 129)

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

List down 2 major benefits of virtual memory

Answer:- (Page 186)

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.

What are the possible system for the input redirection in the UNIX/LINX system

Answer:- (Page 55)

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`

What is the purpose of “stub” in dynamic linking, give answer with respect to memory?

Answer:- (Page 155)

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

Dynamic linking

Answer:- (Page 156)

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.

What is use of mounting in file system?

Answer:- (Page 226)

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.



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

How operating attacks the "no preemption" condition necessary for feedback in order to solve the problem of deadlock?

Answer:- (Page 129)

No preemption:

Resources cannot be preempted. That is, after using it a process releases a resource only voluntarily.

What is pager? Give answer with respect to virtual memory.

Answer:- (Page 187)

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.

What does the following command do in the LINUX/UNIX operating system?

Answer:- (Page 26)

```
$mkdir ~/courses/cs604/program
```

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

How you can differentiate between external and internal fragmentation?

Answer:- [click here for details](#)

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

Three major frame allocation schemes?

Answer:- (Page 205)

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



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

Consider the round robin technique .do you think that the deadlock or starvation can happen in the round robin tech scheduling

Answer:- (Page 86)

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 timesharing 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.

Explain the work of copy on write with respect to virtual memory.

Answer:- (Page 194)

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.

Context switching

Answer:- (Page 31)

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.

Formula to find size of page table.

Answer:- (Page 166)

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

File control block

Answer:- (Page 233)

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).

Disk drives, meeting this responsibility entail having a fast access time and disk bandwidth. Marks 3:

Answer:- (Page 243)

i. structure of 2-level page table,

ii. If a process exits but its threads are still running, will they continue?



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

One advantage and one disadvantage of using a large block size to store file data (NET),

Answer:- (Page)

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 .

Three types of access modes and classes of users in UNIX protection, (P # 230)

Answer:- (Page 230)

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

Possible criteria to decide that which process should be terminated while dead lock detection and recovery.

OR

How can we recover from deadlocks? 2 marks

Answer:- (Page 149)

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

What is mounting in the file system? And where is the mount point?

OR

What is mounting? And what is Mount Point? (P#226)

Answer:- (Page 226)

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.



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

Write one advantage and one disadvantage of using large size block.

Answer:- [click here for details](#)

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.

Write the formula to calculate page table size

Answer:- (Page 166)

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 lfl based on our discussion so far).

Name two registers used for segmentation and what they represent?(3 marks)

OR

Name of two registers used in segmentation?

Answer:- (Page 176)

Segment-table base register (STBR)

Segment-table length register (STLR) indicates

How to detect cycles in acyclic graph?

Answer:- (Page 224)

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.

Name three access modes use in file protection?

Answer:- (Page 178)

1. Read
2. Write
3. Execute

How semaphore algorithm works in n-critical problem 5marks

Answer:- (Page 109)

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 P_i is organized as follows:
do

```
{  
wait(mutex);  
Critical section
```



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

```
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

Some way to reduce external fragmentation 2marks

Answer:- (Page 235)

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.

Four characteristic of deadlock prevention 2 marks

Answer:- (Page 133,134)

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

Is starvation and deadlock are same. Accept or reject with solid reason 3 marks

Answer:- (Page 113,120)

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.

Why we need medium term scheduling? 3marks

Answer:- (Page 34)

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

Difference between deadlock avoidance and deadlock (snowz) prevention? 3marks

Answer:- (Page 133)

1. 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.



(Subjective-CS604 OPERATING SYSTEM) **(From Final Term Papers “No repeated Questions”) June 2014**

2. **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.

Source open software are helpful for testing algorithms as compared to pirated software?

Answer:- [click here for details](#)

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

Hardware required for swapping, paging and demand paging?

Answer:- (Page 158,162,186)

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.

Semaphores Algorithm?

Answer:- (Page 107)

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];



(Subjective-CS604 OPERATING SYSTEM)

(From Final Term Papers “No repeated Questions”) June 2014

```
boolean lock;
The structure of process Pi is:
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 P_i can 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.

Who generate physical address and logical address?

Answer:- (Page 253)

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



(Subjective-CS604 OPERATING SYSTEM)

(From Final Term Papers “No repeated Questions”) June 2014

Write the names of common file structures.

Answer:- (Page 125)

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

Write the method through which Linus/Unix commands can communicate with each other.

Answer:- (Page 26)

Answer \$mkdir ~/courses/cs604/program

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

Possible criteria to decide that which process should be terminated

Answer:- (Page 149)

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.

Sometimes mkfifo call may be failure,write the reasons for failure if mkfifo call in Unix?

Answer:- (Page 57)

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()

Which factors are determine to choose a process for Termination?

Answer:- (Page 148)

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



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

- 11) How many processes will need to be terminated
- 12) Whether the process is interactive or batch

How can you calculate size of page table, give formula?

Answer:- (Page 165)

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 lfl based on our discussion so far). Page table size = 16 * 5 bits

A code was given and question was to tell whether it satisfies mutual exclusion?

Answer:- [click here for details](#)

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.

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:- [click here for details](#)

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



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

Identify the necessary information that must be stored in process control block during the execution of program.

Answer:- (Page 30)

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.

Name the steps involved in converting the source code into executable program?

Answer:- (Page 154)

An instruction cycle sometimes called fetch-and-execute cycle, fetch-decode-execute cycle, or FDX) is the basic operation cycle of a computer.

Translation of a source program in a high-level or assembly language involves compilation and linking of the program. 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
```

Do you agree using Pre-paging help to prevent thrashing? 2 Marks

Answer:- (Page 214)

Yes I am agreed, Pre-paging helps to prevent thrashing Pre-paging is an attempt to prevent this high level of initial paging. The strategy is to bring into memory at one time all the pages that will be needed.



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

What is the major purpose of Safety algorithm in Banker’s algorithm? 2 Marks

Answer:- (Page 143)

The Banker's algorithm is a resource allocation and deadlock avoidance algorithm that tests for safety by simulating the allocation of predetermined maximum possible amounts of all resources, and then makes an "s-state" check to test for possible deadlock conditions for all other pending activities, before deciding whether allocation should be allowed to continue.

In other word Banker’s algorithm executes the Safety algorithm to determine if the resultant system will be in a safe state or not, if the resultant system will be in a safe state then Banker’s algorithm allocates the resource to that process.

"Virtual memory is an extremely large main memory” do you agree with the given statement or not?

Give reasons to support you answer 3

Answer:- [click here for details](#)

This is a user point of view that Virtual memory is extremely large main memory. Actually Virtual memory increases the available memory your computer has by enlarging the "address space," or places in memory where data can be stored. It does this by using hard disk space for additional memory allocation. However, since the hard drive is much slower than the RAM, data stored in virtual memory must be mapped back to real memory in order to be used.

The process of mapping data back and forth between the hard drive and the RAM takes longer than accessing it directly from the memory. This means that the more virtual memory is used, the more it will slow your computer down. While virtual memory enables your computer to run more programs than it could otherwise, it is best to have as much physical memory as possible. This allows your computer to run most programs directly from the RAM, avoiding the need to use virtual memory. Having more RAM means your computer works less, making it a faster, happier machine.

When due to page fault CPU utilization and other IO devices utilization decreases and disk utilization increase. Which process may cause this condition and how is responsible to overcome this situation?

5 Marks

Answer:- (Page 208)

If a process does not have “enough” pages, the page-fault rate is very high. This leads to low CPU utilization. The operating system thinks that it needs to increase the degree of multiprogramming, because it monitors CPU utilization and find it to be decreasing due to page faults. Thus another process is added to the system and hence thrashing occurs and causes throughput to plunge.

A process is thrashing if it is spending more time paging (i.e., swapping pages in and out) than executing.

Thrashing results in severe performance problems:

Low CPU utilization

High disk utilization

Low utilization of other I/O devices



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

While implement wait and signal operations implemented on two process P0 and P1, while the sequence of wait and signal operation that can cause these process to be caught into the problem of mutual exclusive 3

OR

Perform semaphore using P0 and P1 for starvation. 3Marks

Answer:- (Page 116)

Here is an example of starvation. The code structures are self-explanatory.

P0	P1
wait(S);	wait(S);
...	...
wait(S);	signal(S);
...	...

In address spaces of 32-bit and above machines, you are required to identify and briefly explain two paging techniques that can reduce the size of page table in large logical address space. 5 marks

Answer:- (Page 173)

In the 32-bit machine we need to partition p into two parts, p1 and p2. p1 is used to index the outer page table and p2 to index the inner page table. Thus the logical address is divided into a page number consisting of 20 bits and a page offset of 12bits. Since we page the page table, the page number is further divided into a 10-bit page number, and a 10-bit page offset. This is known as **two-level paging**.

Write the names of common file structures. (2)

Answer:- (Page 217)

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

None

File is a sequence of words, bytes

Simple record structure

Lines

Fixed length

Variable length

Complex Structures

Formatted document

Relocatable load file



(Subjective-CS604 OPERATING SYSTEM)

(From Final Term Papers “No repeated Questions”) June 2014

Differentiate between dead lock avoidance and dead lock prevention?

Answer:- [click here for details](#)

The Difference between Deadlock Prevention and Deadlock Avoidance

- **Deadlock Prevention:**
 - 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.
- **Deadlock Avoidance:**
 - The system dynamically considers every request and decides whether it is safe to grant it at this point,
 - The system requires additional a priori information regarding the overall potential use of each resource for each process.
 - Allows more concurrency.

Similar to the difference between a traffic light and a police officer directing traffic.

Identify the necessary information that must be stored in process control block during the execution of program. (5)

Answer:- (Page 33)

Each process is represented in the operating system by a process control block (PCB) – also called a task control block. A PCB contains many pieces of information associated with a specific process, including these:

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.



(Subjective-CS604 OPERATING SYSTEM) **(From Final Term Papers “No repeated Questions”) June 2014**

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.

Difference between local & global replacement with respect to page replacement algorithm.

Answer:- [\(Page 207\)](#)

Local vs Global Replacement

If process P generates a page fault, page can be selected in two ways:

Select for replacement one of its frames.

Select for replacement a frame from a process with lower priority number.

Global replacement allows a process to select a replacement frame from the set of all frames, even if that frame belongs to some other process; one process can take a frame from another. Local replacement requires that each process select from only its allocated frames.

What is stack?

Answer:- [click here for details](#)

LIFO stacks, also known as "push down" stacks, are the conceptually simplest way of saving information in a temporary storage location for such common computer operations as mathematical expression evaluation and recursive subroutine calling.

What is meant by sequential search?

Answer:- [click here for details](#)

In computer science, linear search or sequential search is a method for finding a particular value in a list, which consists of checking every one of its elements, one at a time and in sequence, until the desired one is found.

What is meant by Sequential Access?

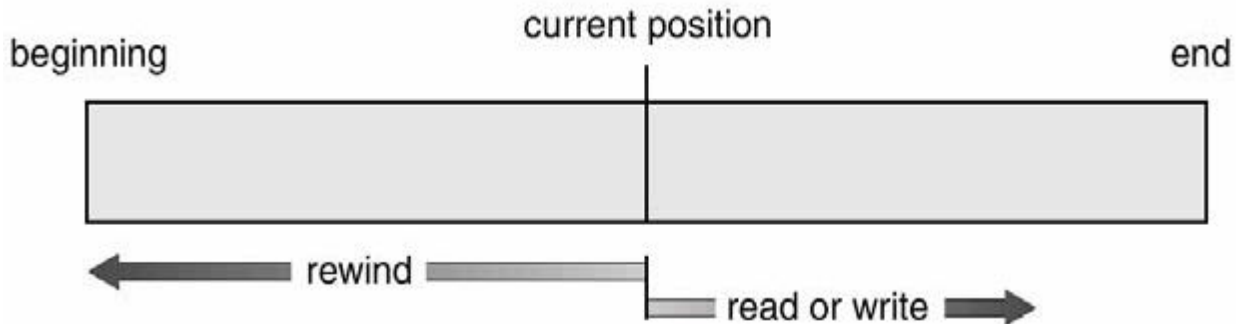
Answer:- [\(Page 221\)](#)

Information in the file is processed in order, one record after the other. A read operation reads the next portion of the file and automatically advances a file pointer which tracks the I/O location. Similarly, a write operation appends to the end of the file and advances to the end of the newly written material. Such a file



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

can be rest to the beginning and on some systems; a program may be able to skip forward or backward, n records.



Segment table base register and segment table length register?

Answer:- (Page 178)

There are registers, relevant to the concept of segmentation.

Segment-table base register (STBR) :

Points to the segment table's location in memory.

Segment-table length register (STLR):

Indicates number of segments used by a program

How can you define compile time? 2

Answer:- (Page 154)

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.

What are the conditions in resource allocation graph by which we can say that the set of processors may or may not be in deadlock? 2

Answer:- (Page 133)

If the graph contains cycles then:

If only one instance per resource type, then a deadlock exists.

If several instances per resource type, **possibility** of deadlock exists.

Which anomaly is involved in FIFO page replacement? 2

Answer:- (Page 201)

FIFO page replacement suffers from Belady's anomaly.



(Subjective-CS604 OPERATING SYSTEM) (From Final Term Papers “No repeated Questions”) June 2014

If a process is starved then it means that the system is in deadlock. Is this statement correct or incorrect? Give your answer with valid arguments. 3

Answer:- [click here for details](#)

A set of processes is deadlocked if each process in the set is waiting for an event that only another process in the set can cause.

Notice that it is possible for a single process to become deadlocked if it is waiting for an event that only it can cause. However, it is usually the case that one process is waiting for a resource that is held by another process. If the other process is itself waiting for a resource held by the first then deadlock occurs.

Note that deadlock is different from starvation. Starvation is the problem that occurs when a process is waiting for a resource that is allocated to other processes, released, and then allocated again to some process other than the one that is starving.

What steps are needed for page replacement?

Answer:- (Page 199)

1. Find the location of the desired 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.

Define mounting in UNIX n command...?

Answer:- (Page 229)

All files accessible in a Unix system are arranged in one big tree, the file hierarchy, rooted at /. These files can be spread out over several devices. The mount command serves to attach the file system found on some device to the big file tree. Conversely, the umount command will detach it again. Here is the syntax of the mount command

mount -t type device dir

This command tells the kernel to attach the file system found on device (which is of type type) at the directory dir. The previous contents (if any) and owner and mode of dir become invisible. As long as this file system remains mounted, the pathname dir refers to the root of the file system on device.

Disk scheduling SCAN and Look (Page#247)

Answer:- (Page 248)

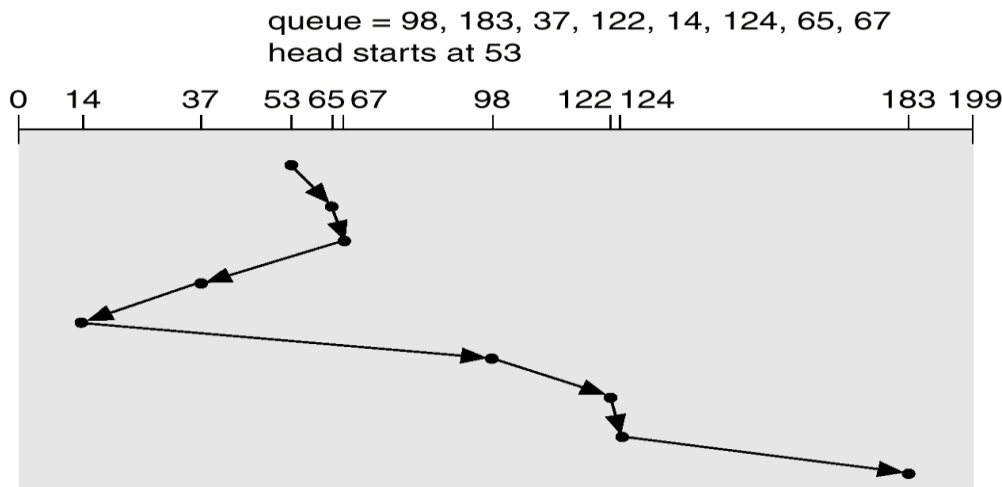
Scan

In the Scan algorithm the disk arm starts at one end of the disk, and moves toward the other end, servicing requests as it reaches each cylinder, until it gets to the other end of the disk. At the other end, the direction of head movement is reversed and servicing continues. The head continuously scans back and forth across the disk.



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

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.

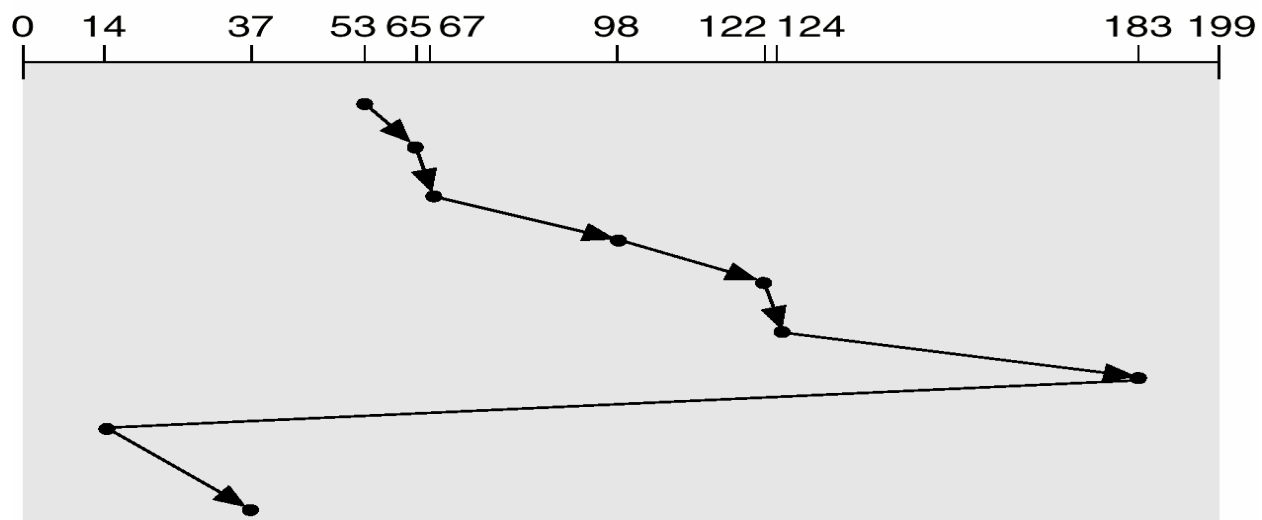


Scan disk scheduling algorithm with disk head moving from right to left

Look algorithm

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. For the given request queue, the total head movement (seek distance) for the Look algorithm is 208.

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53



Look disk scheduling algorithm with the disk head moving from right to left



(Subjective-CS604 OPERATING SYSTEM)

(From Final Term Papers “No repeated Questions”) June 2014

Safe and unsafe state, safe sequence ?

Answer:- (Page 137)

Safe State

A state is safe if the system can allocate resources to each process in some order and still avoid a deadlock.

Unsafe State

A state is unsafe if the system cannot allocate resources to each process in some order and also cannot avoid a deadlock

Safe sequence

We say that a system is in a safe state if all of the processes in the system can be executed to termination in some order; the order of process termination is called safe sequence.

Hold and wait condition for Deadlocks

Answer:- (Page 131)

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

Difference between Bounded waiting and progress condition.

Answer:- (Page 101)

Progress Condition

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.

Real time system definition

Answer:- (Page 10)

A real time system has well defined, fixed time constraints, and if the system does not produce output for an input within the time constraints, the system will fail. For instance, it would not do for a robot arm to be instructed to halt after it had smashed into the car it was building.

Issue with segmentation and solution.

Answer:- (Page 180)

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



(Subjective-CS604 OPERATING SYSTEM) (From Final Term Papers “No repeated Questions”) June 2014

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.

Problems in round robin with swapping.

Answer:- (Page 160)

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 never to swap out processes with pending I/O or to execute I/O in kernel space.

Methods for process communication

Answer:- (Page 46)

- Direct or indirect communication
- Symmetric or asymmetric communication
- Automatic or explicit buffering
- Send by copy or send by reference
- Fixed size or variable size messages

Mmap() command

Answer:- (Page 198)

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.

File I/O with mmap()

```
filides = open(...)
```

```
address = mmap((caddr_t) 0, len, (PROT_READ | PROT_WRITE), MAP_PRIVATE, filides, offset);
```

```
/* use data at address */
```



(Subjective-CS604 OPERATING SYSTEM)

(From Final Term Papers “No repeated Questions”) June 2014

What are the three stages/times when the address is bound to instructions and data?

Answer:- (page 154)

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

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.

What is Load Time? (Page#226)

Answer:- (Page 154)

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.

How can you display the status of suspended and background processes in Unix/Linux

Answer:- (Page 68)

We can use the jobs command to display the status of suspended and background processes.

Moving a process into foreground

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]

where, job_id is the job ID (not process ID) of the suspended or background process. If %job_id is omitted, the current job is assumed.

Moving a process into background

You can use the bg command to put the current or a suspended process into the background. Here is the syntax of the command.

bg [%job_id]

If %job_id is omitted the current job is assumed.

Consider a logical address space of eight pages of 1024 words each, mapped onto 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:- (Page 168)



(Subjective-CS604 OPERATING SYSTEM)

(From Final Term Papers “No repeated Questions”) June 2014

No. of bits needed for p = ceiling $[\log_2 8]$ bits = 3 bits
No. of bits needed for f = ceiling $[\log_2 32]$ bits = 5 bits
No. of bits needed for d = ceiling $[\log_2 2048]$ bits = 11 bits
Logical address size = $|p| + |d| = 3+11$ bits = 14 bits
Physical address size = $|f| + |d| = 5+11$ bits = 16 bits

Drawbacks of semaphore

Answer:- [click here for details](#)

- Simple algorithms require more than one semaphore.
- This increases the complexity of semaphore solutions to such algorithms.
- Semaphores are too low level.
- It is easy to make programming mistakes (e.g. P(s) followed by P(s)).
- The programmer must keep track of all calls to wait and to signal the semaphore.
- If this is not done in the correct order, programmer error can cause deadlock.
- Semaphores are used for both condition synchronization and mutual exclusion.
- These are distinct and different events, and it is difficult to know which meaning any given semaphore may have.

Address Generated by CPU ?

Answer:- (Page 155)

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

Differ between physical address and virtual address?

Answer:- [click here for details](#)

Physical addressing means that your program actually knows the real layout of RAM. When you access a variable at address 0x8746b3, that's where it's really stored in the physical RAM chips.

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**.

With virtual addressing, all application memory accesses go to a page table, which then maps from the virtual to the physical address. So every application has its own "private" address space, and no program can read or write to another program's memory. This is called *segmentation*.

Define and briefly describe, what is memory mapping system calls? 3calls

Answer:- (Page 198)

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.



(Subjective-CS604 OPERATING SYSTEM)

(From Final Term Papers “No repeated Questions”) June 2014

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.

File I/O with mmap()

```
files = open(...)  
address = mmap((caddr_t) 0, len, (PROT_READ | PROT_WRITE), MAP_PRIVATE, files, offset);  
/* use data at address */
```

Disadvantage of dynamic loading?

Answer:- [\(Page 157\)](#)

Run time activity involved in dynamic loading is a disadvantage.

Look algorithm?

Answer:- [\(Page 249\)](#)

This algorithm is a version of SCAN. In this algorithm the arm only goes as far as the last request in each direction, and 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.

Degree of multiprogramming when increased what effect is on CPU utilization?

Answer:- [\(click here for details ,Page 208\)](#)

Multi-programming is affecting CPU utilization and always increases the CPU utilization because it organizes jobs in such ways where CPU executes one job at a time. Meanwhile the remaining jobs kept in memory by operating system. As all the jobs can fit in the memory due to limited memory size availability so jobs in bulk remain on disk and later on they would be loaded to memory for execution.

If a process does not have “enough” pages, the page-fault rate is very high. This leads to low CPU utilization. The operating system thinks that it needs to increase the degree of multiprogramming, because it monitors CPU utilization and find it to be decreasing due to page faults. Thus another process is added to the system and hence thrashing occurs and causes throughput to plunge.

If hard real-time system cannot manage output within given slot then what happen? 3marks

Answer:- [\(Page 10\)](#)

A real time system has well defined, fixed time constraints, and if the system does not produce output for an input within the time constraints, the system will fail. For instance, it would not do for a robot arm to be instructed to halt after it had smashed into the car it was building.

Is disk I/O to swap space generally faster than file system? Give solid reasons (5)

Answer:- [\(Page \)](#)



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

Disk I/O to swap space is generally faster than that to the file system. It is faster because swap space is allocated in much larger blocks, and file lookups and indirect allocation methods are not used. It is therefore possible for the system to gain better paging throughput by copying an entire file image into the swap space at process startup, and then performing demand paging from the swap space.

Vfork() works same as fork() with copy-on-write? (3)

Answer:- (Page)

vfork() operates differently than fork() with copy on write. With vfork() the parent process is suspended and the child process uses the address space of the parent.

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? (2)

Answer:- (Page 181)

Number of pages logical address = $\text{ceiling}[3921/1024] = 4$ (i.e., page number 3)

Consider a process A with segment size is 14, having 5098 bytes. The process generates a logical address (14, 3922). How many pages does the segment have? Mention all steps related to calculation.

Answer:- (Page)

Number of pages the segment have = $\text{ceiling}[5098/1024] = 5$

Consider a process B generates a logical address (15, 3922) whose page size is of 1KB. What will be the value of segment offset d and the physical address if page number 2 is in frame 10? Mention all necessary steps related to calculation.

Answer:- (Page)

Offset (d') = $3922 - 2 * 1K = 1874$

Physical address(p) = $10 * 1K + 1874 = 12114$

One of the responsibilities of O.S is to use computer hardware efficiently, so look Algorithm for disk scheduling,

OR

One of the responsibilities of the operating system is to use the computer system hardware efficiently. To meet this responsibility, we require a disk drive with fast access time and more disk bandwidth.



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

What do you think that how can we improve these two parameter values to make disk drive servicing efficient? Also identify different methods to improve these parameters.

Answer:- (Page 245)

We can improve both the access time and the bandwidth by scheduling the servicing of disk I/O requests in a good order.

Some of the popular disk-scheduling algorithms are:

- First-come-first-serve (FCFS)
- Shortest seek time first (SSTF)
- Scan
- Look
- Circular scan (C-Scan)
- Circular look (C-Look)

Consider the following snapshot of a system:

	Allocation				Max				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P0	0	0	1	2	0	0	1	2	1	5	2	0
P1	1	0	0	0	1	7	5	0				
P2	1	3	5	4	2	3	5	6				
P3	0	6	3	2	0	6	5	2				

From the above data how can you calculate matrix “need” values?
 Calculate need matrix for above given data using banker’s algorithm.

Answer:- (Page 139)

The values of Need Matrix can be calculated by this formula..... Max- Allocation

	Max				Allocation			
	A	B	C	D	A	B	C	D
P0	0	0	1	2	0	0	1	2
P1	1	7	5	0	1	0	0	0
P2	2	3	5	6	1	3	5	4
P3	0	6	5	2	0	6	3	2

	Need			
	A	B	C	D
P0	0	0	0	0
P1	0	7	5	0
P2	1	0	0	2
P3	0	0	2	0



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

Consider the following table to compare the paging with segmentation. Fill your answer with yes and no.

Answer:- [click here for details](#)

Consideration	Paging	Segmentation
Need the programmer be aware that this technique is being used?	NO	Yes
Can the total address space exceed the size of physical memory	Yes	Yes
Can procedures and data be distinguished and separately protected?	NO	Yes
Can tables whose size fluctuates be accommodated easily?	No	Yes
Is sharing of procedures between users facilitated?	No	Yes
How many linear address spaces are there?	1	Many
Why was this technique invented?	To get a large linear address space without having to buy more physical memory	To allow programs and data to be broken up into logically independent address spaces and to aid sharing and protection

Methods for Handling Deadlocks

Answer:-

Principally, we can deal with the deadlock problem in one of three ways:

- We can use a protocol to prevent or avoid deadlocks, ensuring that the system will never enter a deadlock state.
- We can allow the system to enter a deadlock state, detect it, and recover.
- We can ignore the problem altogether and pretend that deadlocks never occur in the system.

Segmentation

Answer:-

- Segment is a logical unit
- Programmer is aware of and uses as a logical unit
- E.g., Code, Heap, Stack



(Subjective-CS604 OPERATING SYSTEM) (From Final Term Papers “No repeated Questions”) June 2014

- Advantages:

- Easy to handle dynamically expanded/shrunked data structures
- Linking is simplified if each procedure occupies a separate segment
- Facilitating sharing procedures/data
- Different protection for different segments

In deadlock prevention strategy do you think it is necessary to check that either safe state exists or not? Give reason to support your answer.

Answer:- (Page 136)

No, it is not necessary to check that either safe state exists or not in deadlock prevention strategy. Because this is done in Deadlock Avoidance.

In prevention strategy we apply Mutual exclusion, Hold and Wait, No preemption and Circular Wait.

Analyze that how an operating system protects the CPU

Answer :- (Page 14)

We must ensure that the operating system maintains control. We must prevent the user program from getting stuck in an infinite loop or not calling system services and never returning control to the CPU. To accomplish this we can use a **timer**, which interrupts the CPU after specified period to ensure that the operating system maintains control. The timer period may be variable or fixed. A *fixed-rate clock* and a *counter* are used to implement a variable timer. The OS initializes the counter with a positive value. The counter is decremented every clock tick by the clock interrupt service routine. When the counter reaches the value 0, a timer interrupt is generated that transfers control from the current process to the next scheduled process. Thus we can use the timer to prevent a program from running too long. In the most straight forward case, the timer could be set to interrupt every N milliseconds, where N is the **time slice** that each process is allowed to execute before the next process gets control of the CPU. The OS is invoked at the end of each time slice to perform various housekeeping tasks.

The main purpose of the computer system is to run different programs, why we run these programs?

Answer :- [click here for details](#)

The main purpose of the computer system is to run different programs, OS run different programs in succession to speed up processing. Operating systems did not exist in their modern and more complex forms.



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

Hardware features were added, that enabled use of runtime libraries, interrupts, and parallel processing. When personal computers became popular in the 1980s, operating systems were made for them similar in concept to those used on larger computers.

The problem with using an acyclic-graph structure is ensuring that there are no cycles. What is the solution?

Answer :- (Page 225)

A file may now have multiple absolute path names. This problem is similar to the aliasing problem in programming languages. Consequently distinct file name may refer to the same files. If we are traversing the entire file system-to find a file, to accumulate statistics, etc, this problem becomes significant since we do not want to traverse the shared structures more than once. Another problem involves deletion. If the file is removed when anyone deletes it, we may end up with dangling pointers to the now-nonexistent file.

Another approach is to preserve the file until all references to it are deleted. When a link or a copy of the directory entry is established, a new entry is added to the file-reference list. When a link is deleted, we remove its entry on the list. The file is deleted when its file-reference list is empty. Since the reference list can be very large we can keep a count of the number of references. A new link or directory increments the **reference count**, deleting a link or entry decrements the count. When the count is 0, the file can be deleted. UNIX uses this solution for hard links. **Backpointers** can also be maintained so we can delete all pointers.

Which anomaly is involved in FIFO page replacement?

Answer :- (Page 202)

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 frames increases, whereas we would expect that giving more memory to a process would improve its performance.

Why overlays and virtual memory are identical? (2)

OR

Why Overlays memory management technique is considered to be related with Virtual Memory?

Answer :- (Page 188)



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

To enable a process to be larger than the amount of memory allocated to it, we can use overlays. In Virtual Memory, A program would no longer be constrained by the amount of physical memory that is available. Users would be able to write programs for an extremely large virtual address space simplifying the programming task.

As overlays are doing something like Virtual Memory so Overlays memory management technique is considered to be related with Virtual Memory.

What will happen if `exit ()` is called instead of `pthread_exit ()` in multithreaded system?

Answer :-

`pthread_exit` is used to explicitly exit a thread. Typically, the `pthread_exit()` routine is called after a thread has completed its work and is no longer required to exist. `pthread_exit()` is called from the thread itself to terminate its execution (and return a result) early.

`exit ()` is called instead of `pthread_exit ()` in multithreaded system then whole program as `exit()`, and the shutdown hooks (if properly set up) take care of doing all necessary shutdown ceremonies such as closing files, releasing resources etc.

If it is not possible to find a safe sequence after running the safety algorithm for a requesting process then what does it means? Should the request for that process be fulfilled?

Answer :- (Page 141)

If it is not possible to find a safe sequence after running the safety algorithm for a requesting process then it means that system may face dead lock by going in unsafe state and we should not fulfill the request for that process.

Let us consider a situation in which each process gets a small unit of CPU time, called time slice or quantum, which is usually 10-100 milliseconds After this time has elapsed, the process is preempted -t- and added to the end of the ready queue. Which scheduling algorithm is best suited for above situation?

Answer :- (Page 88)

The round-robin (RR) scheduling algorithm is best suited for above situation



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

How can we get protection for a shared segment? What if one process only wants other processes to read segment?

Answer :- (Page 180)

Segments are shared when entries in the segment tables of two different processes point to the same physical location. The memory mapping hardware will check the protection bits associated with each segment-table entry to prevent illegal access to memory, such as attempts to write into a read only segment.

The bits associated with each entry in the segment table, for the purpose of protection are:

Validation bit: if the validation bit is 0, it indicates an illegal segment

Read, write, execute bits

Do you feel that is there any main concern about time constraint in real time operating system that a programmer must keep that in mind while writing an Operating System for a real-time environment? If not then give reason to support your answer and if yes then mention that main concern?

Answer :- (Page 10)

A real time system has well defined, fixed time constraints, and if the system does not produce output for an input within the time constraints, the system will fail. For instance, it would not do for a robot arm to be instructed to halt after it had smashed into the car it was building.

Let us consider an example of frame allocation:

Number of free frames = 64

Number of processes = 3

Process sizes: P1= 10 pages; P2 = 40 pages; P3 = 127 pages

Discuss how many free frames will be put in the free frames list by using Proportional Allocation?

Answer :- (Page 208)



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

Proportional Allocation

s_i = Size of process P_i

$S = \sum s_i$

m = Number of free frames

a_i = Allocation for $P_i = (s_i / S) * m$

$a_1 = (10 / 177) * 64 = 3$ frames

$a_2 = (40 / 177) * 64 = 14$ frames

$a_3 = (127 / 177) * 64 = 45$ frames

Two free frames are put in the list of free frames

Different type of threads work in operating system, one of them is process threads and other one is kernel threads. If the kernel of the operating system does not know about these threads and the operating systems is fully aware of them and operating system does not know that these threads use either M:1 or M:N mapping. These threads are scheduled by the thread library and are not associated with any process but every thread belongs to a process and these threads are very easily managed. You need identify the type of these threads.

Answer :- (Page 73)

User threads are supported above kernel and are implemented by a thread library at the user level.

Write down the type of bits associated with each entry of segment table for protection?

Answer :- (Page 178)

The segment table maps the two-dimensional logical addresses to physical addresses.

Each entry of a segment table has a *base* and a *segment limit*. The segment base contains the starting physical address where the segment resides in memory, whereas the segment limit specifies the length of the segment.

There are two more registers, relevant to the concept of segmentation:

Segment-table base register (STBR) points to the segment table's location in memory.

Segment-table length register (STLR) indicates number of segments used by a program



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

How can you 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. Another Definition: Internal Fragmentation is the area in a region or a page that is not used by the job occupying that region or page. This space is unavailable for use by the system until that job is finished and the page or region is released.

"Critical section means the section of code in two processes or more than two processes that is used to update a resource (e.g. a shared variable) which is shared between these processes." —Do you agree with the statement or not? If not, then give reason to support your answer.

Answer :- (Page 100)

Disagree, Because

Critical Section: A piece of code in a cooperating process in which the process may update shared data (variable, file, database, etc.).

Name any two schemes that allow efficient implementation of page table?

Answer :- (Page 173)

Hierarchical / Multilevel Paging

Hashed Page Table

Inverted Page Table



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

Under what conditions can you use the Wait-for graph to detect deadlock?

Answer :- (Page 147)

Single Instance of Each Resource Type

If all resources have only a single instance, then we can define a deadlock detection algorithm that uses a variant of the resource allocation graph, called a **wait-for graph**.

We obtain this graph from the resource allocation graph by removing the nodes of type resource and collapsing the appropriate edges.

What is a file control block?

Answer :- (Page 235)

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).

Write a code or pseudo code by using monitor

i) To deposit money in a bank account

ii) To withdraw money from a bank account.

Answer :-

i) To deposit money in a bank account

```
monitor BankAccount
{
public:
    BankAccount() { balance = 0; }

    void withdraw(int amount) {
        balance -= amount;
    }
    void deposit(int amount) {
        balance += amount;
    }

private:
    int balance;
```



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

```
};
```

ii) To withdraw money from a bank account.

```
monitor BankAccount
{
public:
    BankAccount() { balance = 0; }

    void withdraw(int amount) {
        balance -= amount;
    }
    void deposit(int amount) {
        balance += amount;
    }

private:
    int balance;
};
```

Do you think that layered approach in operating system is better than monolithic approach? Give reasons to support your answer.

Answer :- (Page 21)

Yes, layered approach in operating system is better than monolithic approach because the main advantage of the layered approach is modularity. The layers are selected such that each uses functions and services of only lower layers. This approach simplifies debugging and system verification.

What is the advantage of Dynamic Loading? Give your answer with respect to Main Memory Management.

Answer :- (Page 157)

The advantage of dynamic loading is that an unused routine is never loaded. This means that potentially less time is needed to load a program and less memory space is required.



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

Dynamic programming does not require special support from the operating system.

How does a "system call" help the operating system?

Answer :- (Page 18)

System calls provide the interface between a process and the OS. System call generally available as assembly language instructions. The system call interface layer contains entry point in the kernel code; because all system resources are managed by the kernel any user or application request that involves access to any system resource must be handled by the kernel code, but user process must not be given open access to the kernel code for security reasons. So that user processes can invoke the execution of kernel code, several openings into the kernel code, also called *system calls*, are provided. System calls allow processes and users to manipulate system resources such as files and processes.

System calls can be categorized into the following groups:

Process Control

File Management

Device Management

Information maintenance

Communications

Virtual Memory is the separation of user logical memory from physical memory. State Virtual Memory makes programming either easy or difficult?

Answer :- (Page 188)

Virtual memory makes the task of programming easier because the programmer need not worry about the amount of physical memory, or about what code can be placed in overlays; she can concentrate instead on the problem to be programmed.

In addition to separating logical memory from physical memory, virtual memory also allows files and memory to be shared by several different processes through page sharing.

Out of MVT (Multiprogramming with Variable Tasks) and MFT (Multiprogramming with Fixed Tasks), which one do you think is best suited to cause internal fragmentation and which one is best suited to cause external fragmentation? Also differentiate briefly between MVT and MFT



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

Answer :- (Page 164)

MFT (Multiprogramming with Fixed Tasks): internal fragmentation

In this technique, memory is divided into several fixed-size partitions. Each partition may contain exactly one process. Thus the degree of multiprogramming is bound by the number of partitions. In this multiple partition method, when a partition is free, a process is selected from the input queue and is loaded in the free partition. When the process terminates, the partition becomes available for another process.

MVT (Multiprogramming with Variable Tasks): external fragmentation

This is the generalization of the fixed partition scheme. It is used primarily in a batch environment. This scheme of memory management was first introduced in IBM OS/MVT (multiprogramming with a varying number of tasks).

Do you feel that is there any preemptive version of Shortest-Job-First (STE) available? If yes then elaborate the working and performance (in terms of average waiting time) of that preemptive version as compare to non preemptive version, if not, then give reason to support your answer?

Answer :- (Page 90)

Yes, preemptive version of Shortest-Job-First (STE) is available.

The SJF algorithm may either be preemptive or non-preemptive. When the CPU is available, it is assigned to the process that has the smallest next CPU burst. If two processes have the same length next CPU burst, FCFS scheduling is used to break the tie.

SJF is an optimal algorithm because it decreases the wait times for short processes much more than it increases the wait times for long processes. SJF scheduler is better suited for batch systems, in which minimizing the turnaround time is the main criterion.

If a process does not have "enough" pages, the page-fault rate is very high. This leads to low CPU utilization. The operating system needs to increase the degree of multiprogramming because it monitors the CPU utilization. When another process added, will it increase the throughput of the system?

Answer :- (Page208)



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

If a process does not have “enough” pages, the page-fault rate is very high. This leads to low CPU utilization. The operating system thinks that it needs to increase the degree of multiprogramming, because it monitors CPU utilization and find it to be decreasing due to page faults. Thus another process is added to the system and hence thrashing occurs and causes throughput to plunge.

Do overlays increase a burden on a programmer as compare to virtual memory? Give reason to support your answer.

Answer :- (Page 159)

Yes, overlays increase a burden on a programmer as compare to virtual memory because: We cannot partition all problems into overlays and also we are responsible of writing the overlays driver while on the other side, 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. Virtual memory makes the task of programming easier because the programmer need not worry about the amount of physical memory, or about what code can be placed in overlays.

List any three criterion when the fork() system call fails to complete its task?

Answer :- (Page 41)

The fork() system may fail due to a number of reasons. One reason maybe that the maximum number of processes allowed to execute under one user has exceeded, another could be that the maximum number of processes allowed on the system has exceeded. Yet another reason could be that there is not enough swap space.

Write the names of three commonly used methods for file space allocation?

Answer :- (Page 236)

Here are the three commonly used methods for file space allocation.

Contiguous allocation

Linked allocation

Indexed allocation



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

What does a Page Table entry contain?

Answer :- (Page 164)

Page table entry contains the frame number, f , where page p is loaded.

Preemptive Shortest Job First scheduling algorithm is best algorithm for minimizing the waiting time for the processes. How can you calculate the average waiting time in preemptive Shortest Job First :- scheduling algorithm?

Answer :- (Page 86)

SJF is an optimal algorithm because it decreases the wait times for short processes much more than it increases the wait times for long processes. SJF scheduler is better suited for batch systems, in which minimizing the turnaround time is the main criterion.

When the CPU is available, it is assigned to the process that has the smallest next CPU burst. If two processes have the same length next CPU burst, FCFS scheduling is used to break the tie.

Non-preemptive SJF waiting time:

CPU given time-Arrival time

Preemptive SJF waiting time:

Terminated- Arrival time-CPU burst

Define Indexed Allocation as a Space Allocation Method

Answer :- (Page 238)

Indexed allocation brings all the pointers to the block together into a disk block, known as the **index block**. Here is the logical view of the relationship between an index block and a file's data blocks.

Each file has its own index block, which is an array of disk block addresses. The i th entry in the index block points to the i th block of the file. The directory contains the address of the index block. To read the i th block, we use the pointer in the i th index-block entry to find and read the desired block.



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

When a process is rolled out of memory, it loses its ability to use the CPU (at least for a while). Describe another situation where a process loses its ability to use the CPU, but where the process does not get rolled out.

Answer :- (Page 14)

When an interrupt occurs the process loses the CPU, but regains it as soon as the handler completes. The process is never rolled out of memory.

How we do memory protection in memory management?

Answer:- (Page 10)

Using two CPU registers, specifically designed for this purpose, can provide memory protection. These registers are:

Base register - it holds the smallest legal physical memory address for a process

Limit register - it contains the size of the process

How can you achieve memory protection in paging?

Answer:- (Page 172)

Memory protection in paging is achieved by associating protection bits with each page.

These bits are associated with each page table entry and specify protection on the corresponding page.

The primary protection scheme guards against a process trying to access a page that does not belong to its address space. This is achieved by using a valid/invalid (v) bit. This bit indicates whether the page is in the process address space or not. If the bit is set to invalid, it indicates that the page is not in the process's logical address space. Illegal addresses are trapped by using the valid-invalid bit and control is passed to the operating system for appropriate action.

There is a technique that provides the solution to external fragmentation but it may happen that internal fragmentation is still an issue. Suggest the name of that technique and how this technique makes it possible to avoid fragmentation issues?

Answer:- (Page 164)

Multiprogramming with Variable Tasks (MVT) is a technique that provides the solution to external fragmentation but it may happen that internal fragmentation is still an issue.

Eliminates internal fragmentation:

Find a region the exact right size (leave a hole for the remainder).

Not quite true, can't get a piece with 10A755 bytes. Would get say 10A760.

But internal fragmentation is *much* reduced compared to MFT. Indeed, we say that internal fragmentation has been eliminated.



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

Since disk space is limited so we need to reuse the space from deleted files for new files if possible. How system keeps tracking of free disk space?

Answer:- (Page 242)

To keep track of free disk space, the system maintains a free-space list. The free space list records all free disk blocks-those not allocated to some file or directory. To create a file we search the free-space list for the required amount of space and allocate the space to the new file. This space is then removed from the free-space list. When a file is deleted, its disk space is added to the free space list.

Dining philosopher problem deadlocks that occur while eating can be removed by applying certain strategies. Suggest any two of them.

Answer:- (Page 123)

Allow at most four philosophers to be sitting simultaneously at the table.

Allow a philosopher to pick up her chopsticks only if both chopsticks are available (to do this she must pick them up in a critical section)

Use an asymmetric solution; that is, an odd philosopher picks up first her left chopstick, whereas an even philosopher picks up her right chopstick and then her left chopstick.

Consider the following programming techniques and data structures. In demand paged environment, which of the following is/are good or bad? Explain your answer.

1. Stack
2. Hash table
3. Sequential search

Answer:- [click here for details](#)

- Stack -- good, stack operations are local
- Hashed Symbol Table -- not good, operations are not local
- Sequential Search -- good
- Binary Search -- not good, unless the table fits in few pages
- Pure Code -- good, sequential access
- Vector Operations -- good, sequential access
- Indirections -- not good, contains jumps

Name two differences between logical and physical addresses.

Answer:- (Page)

Logical address is an address seen by the CPU while a physical address is seen by the memory.



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

A physical address is limited to the amount of installed memory while a logical address is limited by the address size of the processor.

Describe the following allocation algorithms:

Answer:- (Page)

- (A) First-fit: search the list of available memory and allocate the first block that is big enough.
- (B) Best-fit: search the entire list of available memory and allocate the smallest block that is big enough.
- (C) Worst-fit: search the entire list of available memory and allocate the largest block.

Consider a logical address space of eight pages of 1024 words each, mapped onto 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 :-

- A. Logical address: 13 bits
- B. Physical address: 15 bits

Consider the following segment table:

Segment	Base	Length
0	219	600
1	2300	14
2	90	100
3	1327	580
4	1952	96

What are the physical addresses for the following logical addresses?

a. 0,430 b. 1,10 c. 2,500 d. 3,400 e. 4,112

Answer:-

- A. $219 + 430 = 649$
- B. $2300 + 10 = 2310$
- C. illegal reference, trap to operating system
- D. $1327 + 400 = 1727$
- E. illegal reference, trap to operating system



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

Consider a demand paging system with the following time measured utilizations:

CPU: 20 %
Paging Disk: 97,7 %
Other I/O Devices: 5 %

Which of the following will probably improve CPU utilization? Explain your answers.

1. Install a faster CPU.
2. Install a bigger paging disk.
3. Increase the number of processes
4. Decrease the number of processes
5. Install more main memory.
6. Install a faster paging disk, or multiple controllers with multiple hard disks.
7. Add pre-paging to the page fetch algorithms.
8. Increase the page size.

Answer:-

1. Install a faster CPU -- No, CPU is waiting most of the time
2. Install a bigger paging disk -- No, it is not the problem
3. Increase the number of processes -- Never, will increase thrashing
4. Decrease the number of processes -- Best Idea
5. Install more main memory -- Why not? can hold more pages in memory and thus, obtain less page faults
6. Install a faster paging disk, or multiple controllers with multiple hard disk -- Yes, as the disk is the bottleneck, the CPU gets data more quickly.
7. Add pre-paging to the page fetch algorithms -- Could help, above all, if programs follow the locality principle.
8. Increase the page size -- Will reduce the number of page faults if programs follow the locality principle. If not, it could result in higher paging activity because fewer pages can be kept in main memory and more data needs to be transferred per page fault.

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:- (Page)

Given memory partitions of 100K, 500K, 200K, 300K, and 600K (in order), how would each of the First-fit, Best-fit, and Worst-fit algorithms place processes of 212K, 417K, 112K, and 426K (in order)? Which algorithm makes the most efficient use of memory?

Answer:-



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

a. First-fit:

212K is put in 500K partition

417K is put in 600K partition

112K is put in 288K partition (new partition $288K = 500K - 212K$)

426K must wait

b. Best-fit:

212K is put in 300K partition

417K is put in 500K partition

112K is put in 200K partition

426K is put in 600K partition

c. Worst-fit:

212K is put in 600K partition

417K is put in 500K partition

112K is put in 388K partition

426K must wait

In this example, Best-fit turns out to be the best.

Process in normal execution but sometimes its resources are preempted write down the method to handle this condition?

OR

Process Termination

Answer:-

To eliminate deadlocks by aborting a process, we use one of two methods. In both methods, the system reclaims all resources allocated to the terminated processes.

- Abort all deadlocked processes: This method clearly will break the deadlock cycle, but at great expense; the deadlocked processes may have computed for a long time, and the results of these partial computations must be discarded and probably will have to be recomputed later.
- Abort one process at a time until the deadlock cycle is eliminated: This method incurs considerable overhead, since, after each process is aborted, a deadlock-detection algorithm must be invoked to determine whether any processes are still deadlocked.

In Linux OS describe working of copy on write in virtual memory?(5 marks)

Answer:- (Page)

Copy-on-write (sometimes referred to as "COW") is an optimization strategy used in computer programming. Copy-on-write stems from the understanding that when multiple separate tasks use initially identical copies of some information (i.e., data stored in computer memory or disk storage), treating it as



(Subjective-CS604 OPERATING SYSTEM) **(From Final Term Papers “No repeated Questions”) June 2014**

local data that they may occasionally need to modify, then it is not necessary to immediately create separate copies of that information for each task.

Copy-on-write finds its main use in virtual memory operating systems; when a process creates a copy of itself, the pages in memory that might be modified by either the process or its copy are marked copy-on-write. When one process modifies the memory, the operating system's kernel intercepts the operation and copies the memory thus a change in the memory of one process is not visible in another's.

Another use involves the calloc function. This can be implemented by means of having a page of physical memory filled with zeros. When the memory is allocated, all the pages returned refer to the page of zeros and are all marked copy-on-write. This way, the amount of physical memory allocated for the process does not increase until data is written. This is typically done only for larger allocations.

Copy-on-write can be implemented by notifying the MMU that certain pages in the process's address space are read-only. When data is written to these pages, the MMU raises an exception which is handled by the kernel, which allocates new space in physical memory and makes the page being written correspond to that new location in physical memory.

One major advantage of COW is the ability to use memory sparsely. Because the usage of physical memory only increases as data is stored in it, very efficient hash tables can be implemented which only use little more physical memory than is necessary to store the objects they contain. However, such programs run the risk of running out of virtual address space — virtual pages unused by the hash table cannot be used by other parts of the program. The main problem with COW at the kernel level is the complexity it adds, but the concerns are similar to those raised by more basic virtual-memory concerns such as swapping pages to disk; when the kernel writes to pages, it must copy any such pages marked copy-on-write.

What are the problems that occur while trying to violate the condition of hold and wait for deadlock prevention? 2

Answer:-

We've enumerated 4 conditions that must hold for deadlock to occur. So violating any one of them with eliminates the possibility of deadlocking.

Under which condition do you use banker's algorithm to detect deadlock 2

Answer:- (Page 140)

When a process requests a set of resources, the system must determine whether the allocation of these resources will leave the system in a safe state. Then Safety algorithm is used for the given system state to



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

determine if the system is in a safe state. If it will, the resources are allocated; otherwise the process must wait until some other process releases enough resources.

Identify the name which is best suited for a bit that associated with each page in a page table and the purpose of the bit is to indicates whether the page is the process address space or not ?

Answer:- (Page 172)

Memory protection in paging is achieved by associating protection bits with each page. These bits are associated with each page table entry and specify protection on the corresponding page. The primary protection scheme guards against a process trying to access a page that does not belong to its address space. This is achieved by using a valid/invalid (v) bit. This bit indicates whether the page is in the process address space or not. If the bit is set to invalid, it indicates that the page is not in the process's logical address space. Illegal addresses are trapped by using the valid-invalid bit and control is passed to the operating system for appropriate action.

What do you think that why main memory is kept as a volatile memory instead of permanent storage device 3

Answer:-

Volatile memory requires constant power to maintain the stored information. The fastest memory technologies of today are volatile ones (not a universal rule). Since primary storage is required to be very fast, it predominantly uses volatile memory.

What is the difference between swapper and pager?

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.

With regard to main memory and a secondary storage, where to store C source program and where to load it while execution in main memory or secondary storage support your answer with reason?(3 marks)

Answer:-

Secondary storage for storing c program and for execution we ill use main memory.



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

Every program is stored in secondary storage.. until it is unless we execute it.. when we execute our program, it will first have to be brought and loaded into main memory because the operating system is responsible for execution and OS access to our program needs to come into RAM, so execution is done in main memory...

Write the safe sequence of

	Max usage	current usage
p1	9	3
p2	5	1
p3	8	5

Available resources are=4

Answer:-

Safe sequence is <p2,p3,p1>

File sharing method 05 Marks

OR

A team of software engineers working on a project write down the four possible ways to share files between them. (5)

Answer:- (Page 230)

File Sharing

Sharing of files on multi-user systems is desirable. People working on the same project need to share information. For instance: software engineers working on the same project need to share files or directories related to the project. Sharing may be done through

Duplicating files: Make copies of the file and give them to all team members.

This scheme works well if members of the team are to work on these shared files sequentially. If they work on the files simultaneously, the copies become inconsistent and no single copy reflects the work done by all members. However, it is simple to implement.



(Subjective-CS604 OPERATING SYSTEM) (From Final Term Papers “No repeated Questions”) June 2014

Common login for members of a team: The system admin creates a new user group and gives the member access to the new account. All files and directories created by any team member under this account and are owned by the team. This works well if number of teams is small and teams are stable. However a separate account is needed for the current project and the system administrator has to create a new account for every team

Setting appropriate access permissions: Team members put all shared files under one member’s account and the access permissions are set so all the members can access it. This scheme works well if *only* this team’s members form the user group. File access permissions can be changed using the chmod system call:

chmod [options] octal-mode file list

chmod [options] symbolic –mode file-list

A few examples:

–To let people in your UNIX group add, delete, and rename files in a directory of yours - and read or edit other people's files if the file permissions let them – use chmod 775 dirname.

–To make a private file that only you can edit, use chmod 600 filename. To protect it from accidental editing, use chmod 400 filename.

Common groups for members of a team. : System admin creates a new user group consisting of the members of team only. All team members get individual logins and set access permissions for their files so that they are accessible to other group members

Links: A link is a way to establish a connection between the file to be shared and the directory entries of the users who want to have access to this file. The two types of links supported by UNIX:

–Hard link

–Soft/symbolic link

Which component of OS is best suited for secure efficient and orderly manage memory 05M

Answer:-

Memory management unit (MMU) in CPU

Many ways to breakdown deadlock one of them is Preemptive a resource which issue address this technique 03M

Answer:- (Page 153)

Resource Preemption



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

To eliminate deadlocks using resource preemption, we successively preempt some resources from processes and give these to other processes until the deadlock cycle is broken. If preemption is required to deal with deadlocks, then three issues need to be addressed:
Selecting a victim, Rollback, Starvation.

Explain the difference between internal and external fragmentation for segmentation/paging and allocation of files on disk.

Answer:-

Segmentation: Segments are of variable size, the allocation is dynamic. When all blocks of free memory are too small to accommodate a segment, it results in external fragmentation.

Paging: Memory is allocated in fixed-sized unit (page frames), when the request is smaller than page size or can not be evenly divided by page size, at least part of one page (the last page) won't be used by the requester and unavailable for use by others, which is internal fragmentation.

Which of the following increase the CPU utilization select option and give reason?

- a) install a faster CPU
- b) install a bigger paging disk
- c) increase the degree of multiprogramming
- d) decrease the degree of multiprogramming
- e) install more memory
- f) install a faster hard disk
- g) add prepaging to the page-fetch algorithms
- h) increase the page size

Answer:-

The system obviously is spending most of its time paging, indicating over-allocation of memory. If the level of multiprogramming is reduced resident processes would page fault less frequently and the CPU utilization



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

would improve. Another way to improve performance would be to get more physical memory or a faster paging disk.

a. Get a faster CPU - No.

b. Get a bigger paging disk - No.

c. Increase the degree of multiprogramming - No.

d. Decrease the degree of multiprogramming - Yes.

e. Install more main memory - Likely to improve CPU utilization as more pages can remain resident and not require paging to or from the disks.

f. Install a faster hard disk, or multiple controllers with multiple hard disks - Also an improvement, for as the disk bottleneck is removed by faster response and more throughputs to the disks, the CPU will get more data more quickly.

g. Add pre-paging to the page fetch algorithms - Again, the CPU will get more data faster, so it will be more in use. This is only the case if the paging action is amenable to pre-fetching (i.e., some of the access is sequential).

h. Increase the page size - Increasing the page size will result in fewer page faults if data is being accessed sequentially. If data access is more or less random, more paging action could ensue because fewer pages can be kept in memory and more data is transferred per page fault. So this change is as likely to decrease utilization as it is to increase it.

Which term is best suited for edge $P \rightarrow R$ indicating that process p may request resource p some time in the future?(2 marks)

Answer:- (Page 139)

Claim edge to resource allocation graphs. A claim edge $P_i \rightarrow R_j$ indicates that process P_i may request resource R_j at some time in the future. A dashed line is used to represent a claim edge. When P_i requests resource R_j the claim edge is converted to a request edge.

Differentiate between page number (p) and offset (d) ?(2 marks)

Answer:- (Page 168)

Every **logical address** generated by the CPU is divided into two parts: a page number (p) and a page offset (d). The page size is defined by the CPU hardware. If the size of logical address space is 2^m and a page size



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

is 2^n addressing units (bytes or words) , then the high-order $m-n$ bits of a logical address designate the page number and the n low order bits designate offset within the page.

Advantages of partially loaded program (2)

Answer:- (Page 188)

The ability to execute a program that is only partially in memory confers many benefits. A program would no longer be constrained by the amount of physical memory that is available. Users would be able to write programs for an extremely large virtual address space simplifying the programming task.

Because each user program could take less physical memory, more programs could be run at the same time, with a corresponding increase in CPU utilization and throughput with no increase in response time or turnaround time.

Less I/O would be needed to load or swap each user program into memory, so each user program would run faster.

Thus running a program that is not entirely in memory would benefit both the system and the user.

You are required to identify a major issue of segmentation as a result of larger logical address space?

Answer:- (Page 181)

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.

In case of file protection, what should the file owner/creator to be able to control? And what operations need to be controlled?(5 marks)

Answer:- (Page 232)

File owner/creator should be able to control

What can be done?

By whom

Several types of operations may be controlled:

Read: read from the file

Write: write or rewrite to the file

Execute: Load the file into memory and execute it

Append: Write new information at the end of the file

Delete: Delete the file and free its space for possible reuse

List: List the name and attributes of the file



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

If the programmer uses semaphores in wrong places then no problem will occur, do u agree with the statement (2)

Answer:- (Page)

Wrong use of wait and signal operations (in context with semaphores) can cause Mutual Exclusion, Deadlock and Bounded Waiting problem(s).

Does the safe sequence is necessary, (2)

Answer:- (Page 137)

NO, If a system is in a safe state, there can be no deadlocks. An unsafe state is not a deadlocked state; a deadlocked state is conversely an unsafe state. Not all unsafe states are deadlocks, however an unsafe state may lead to a deadlock state. Deadlock avoidance makes sure that a system never enters an unsafe state.

There were few UNIX commands in the directory structures.

Answer:- (Page 222)

The following directory operations are commonly supported in contemporary operating systems. Next to each operation are UNIX system calls or commands for the corresponding operation.

Create — mkdir

Open — opendir

Read — readdir

Rewind — rewinddir

Close — closedir

Delete — rmdir

Change Directory — cd

List — ls

Search

Give at least two types of file structure according to its characteristics?

Answer:- (Page 218)

File Attributes

Every file has certain attributes, which vary from one OS to another, but typically consist of these:

Name: The symbolic file name is the only information kept in human-readable form



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

Type: This information is needed for those systems that support different types.

Location: This location is a pointer to a device and to the location of the file on that device.

Size: The current size of the file (in bytes, words or blocks) and possibly the maximum allowed size are included in this attribute.

Protection: Access control information determines who can do reading , writing, etc.

Owner

Time and date created: useful for security, protection and usage monitoring.

Time and date last updated: useful for security, protection and usage monitoring.

Read/write pointer value

What will be criteria to place the page in the main memory?

Answer:- (Page)

Paged allocation divides the computer's primary memory into fixed-size units called page frames, and the program's virtual address space into pages of the same size.

FIFO Page Replacement

Optimal Algorithm

LRU Page Replacement

Least Frequently Used (LFU)

Most Frequently Used (MFU)

Page Buffering Algorithm

What is page fault, how page fault occur and who is responsible for handling page fault?

Answer:-

A page fault (sometimes #pf or pf) is a trap to the software raised by the hardware when a program accesses a page that is mapped in the virtual address space, but not loaded in physical memory.

If the process tries to access a page that was not brought into memory then access to a page marked invalid causes a page fault trap.

OS is responsible for handling page fault. When the process tries to access locations that are not in memory, the hardware traps the operating system (page fault). The operating system reads the desired into memory and restarts the process as though the page had always been in memory.

Does following producer’s code have any ambiguity? (3)

Do {



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

Wait (full);

Wait (mutex);

...

...

signal (mutex);

signal (empty);

...

...

} while (1);

Answer:- (Page 118)

It should be like it.

```
do {  
...  
produce an item in nextp  
...  
wait(empty);  
wait(mutex);  
...  
add nextp to buffer  
...  
signal(mutex);  
signal(full);  
} while(1);
```

Do you think that FIFO is useful? Provide at least two reasons of justify your answer?

Answer:- (Page)

Named pipe (FIFO): FIFOs (also known as named pipes) are used for communication between related or unrelated processes on a UNIX/Linux system.



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

A named pipe (also called a named FIFO, or just FIFO) is a pipe whose access point is a file kept on the file system. By opening this file for reading, a process gets access to the FIFO for reading. By opening the file for writing, the process gets access to the FIFO for writing. By default, a FIFO is opened for blocking I/O. This means that a process reading from a FIFO blocks until another process writes some data in the FIFO. The same goes the other way around. Unnamed pipes can only be used between processes that have an ancestral relationship. And they are temporary; they need to be created every time and are destroyed when the corresponding processes exit. Named pipes (FIFOs) overcome both of these limitations.

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:- (Page)

P = 12 bits

Offset = 5

Let the number of bits required = x so, $2^x = 4096$ so, $x = 12$

Offset = 5

$2^p = 4096$

$2^p = 2^{12}$

$p = 12$ $2^d = 32$ $2^d = 2^5$

$d = 5$

Maximum no of pages in process address space is one million and total address size is (p + d), process address space is 32-bit with page size 4096 bytes. Calculate no bits for page no (p) and the no of bits required for offset (d)?

Answer:- (Page)

For large address spaces 32bits what paging techniques used for reduce the page table size.

Answer:- (Page)

These techniques reduce the page table size.

Multi-level paging

Hashed page tables

Inverted page table

Is it possible to have a deadlock involving only one process? Explain your answer.

Answer:- (Page)

Recall that there are four necessary conditions for deadlock. Number 4 is circular-wait. Deadlock with one process is not possible, because it is not possible to have circular wait with only one process, thus



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

failing a necessary condition. There is no second process to form a circle with the first one. For the same reason, it is not possible to have a deadlock involving only one resource. This follows directly from the hold-and-wait condition.

Consider the traffic deadlock depicted in the Figure 8.8.

A. Show that the four necessary conditions for deadlock indeed hold in this example.

B. State a simple rule that will avoid deadlocks in this system.

Answer:- (Page

A. The four necessary conditions for deadlock hold in this example for the following reasons

(i) Mutual Exclusion : Each of the vehicles present in the streets hold a non-sharable resource: the part of the road they occupy, which they cannot share with the other vehicles.

(ii) Hold and Wait : Each of the vehicles hold the space resource they occupy and are waiting the space in front of them to be freed by other waiting vehicles.

(iii) No Preemption : There is no possibility of preemption as none of the vehicles can give up their resource. In this situation preemption would have to take the form of a vehicle pulling into a parking lot, or a crane reaching down and lifting a vehicle off the road.

(iv) Circular Wait : Circular wait is present because each vehicle is waiting for the space in front of it, and some vehicles occupy spaces where two vehicles wait on them. It is thus possible to trace a cycle of waiting cars. This is the weakest assertion in the set, though, and is clearly untrue out at the edges somewhere, since some car can clearly move someplace in the city. If you have ever experienced grid-lock, though you know that this is small comfort, and that a rule to avoid even "local" deadlock is extremely desirable.

B. The simple rule that could be used to avoid traffic deadlocks in such a system is that intersections should always remain clear as lights change. In this way, the resource of space in the intersection is freed for use at periodic intervals (light changes).

Assume that you have a page-reference string for a process with m frames (initially all empty). The page-reference string has length p ; n distinct page numbers occur in it. Answer these questions for any page-replacement algorithms:

A. What is a lower bound on the number of page faults?

B. What is an upper bound on the number of page faults?

Answer:-

A. N , because a distinct page numbers always generates a page fault, even if two difference pages refer to the same physical page frame.

B. P , if we reference pages P times, we will generate at most P page faults.



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

Give an example of an application in which data in a file should be accessed in the following order:

Answer:-

- A. Sequentially: A movie player that plays a movie file sequentially.
- B. Randomly: A movie player that allows user to start playing movie at random locations of the file.

Researchers have suggested that, instead of having an access list associated with each file (specifying which users can access the file, and how), we should have a user control list associated with each user (specifying which files a user can access, and how). Discuss the relative merits of these two schemes.

Answer:-

- File control list. Since the access control information is concentrated in one single place, it is easier to change access control information and this requires less space.
- User control list. This requires less overhead when opening a file.

Consider a system where free space is kept in a free-space list.

- A. Suppose that the pointer to the free-space list is lost. Can the system reconstruct the free-space list? Explain your answer.**
- B. Suggest a scheme to ensure that the pointer is never lost as result of memory failure.**

Answer:-

- A. In order to reconstruct the free list, it would be necessary to perform "garbage collection." This would entail searching the entire directory structure to determine which pages are already allocated to jobs. Those remaining unallocated pages could be re-linked as the free-space list.
- B. The free-space list pointer could be stored on the disk, perhaps in several places.

Why must the bit map for file allocation be kept on mass storage, rather than in main memory?

Answer:-

In case of system crash (memory failure) the free-space list would not be lost as it would be if the bit map had been stored in main memory.

Consider a system that supports the strategies of contiguous, linked, and indexed allocation. What criteria should be used in deciding which strategy is best utilized for a particular file?

Answer:-

Contiguous -- if file is usually accessed sequentially, if file is relatively small.



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

Linked -- if file is large and usually accessed sequentially.
Indexed -- if file is large and usually accessed randomly.

State three advantages of placing functionality in a device controller, rather than in the kernel. State three disadvantages.

Answer:-

Three advantages: Bugs are less likely to cause an operating system crash. Performance can be improved by utilizing dedicated hardware and hard-coded algorithms. The kernel is simplified by moving algorithms out of it.

Three disadvantages: Bugs are harder to fix - a new firmware version or new hardware is needed. Improving algorithms likewise require a hardware update rather than just kernel or device driver update. Embedded algorithms could conflict with application's use of the device, causing de-creased performance.

Why might a system use interrupt-driven I/O to manage a single serial port, but polling I/O to manage a front-end processor, such as a terminal concentrator?

Answer:-

Polling can be more efficient than interrupt-driven I/O. This is the case when the I/O is frequent and of short duration. Even though a single serial port will perform I/O relatively infrequently and should thus use interrupts, a collection of serial ports such as those in a terminal concentrator can produce a lot of short I/O operations, and interrupting for each one could create a heavy load on the system. A well-timed polling loop could alleviate that load without wasting many resources through looping with no I/O needed.

None of the disk-scheduling disciplines, except FCFS, is truly fair (starvation may occur).

- A. Explain why this assertion is true.**
- B. Describe a way to modify algorithms such as SCAN to ensure fairness.**
- C. Explain why fairness is an important goal in a time-sharing system.**
- D. Give three or more examples of circumstances in which it is important that the operating system be unfair in serving I/O requests.**

Answer:-

A. New requests for the track over which the head currently resides can theoretically arrive as quickly as these requests are being serviced.

B. All requests older than some predetermined age could be "forced" to the top of the queue, and an associated bit for each could be set to indicate that no new request could be moved ahead of these



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

requests. For SSTF, the rest of the queue would have to be reorganized with respect to the last of these "old" requests.

C. To prevent unusually long response times.

D. Paging and swapping should take priority over user requests. It may be desirable for other kernel-initiated I/O, such as the writing of file system metadata, to take precedence over user I/O. If the kernel supports real-time process priorities, the I/O requests of those processes should be favored.

Suppose that a disk drive has 5000 cylinders, numbered 0 to 4999. The drive is currently serving a request at cylinder 143, and the previous request was at cylinder 125. The queue of pending requests, in FIFO order, is 86, 1470, 913, 1774, 948, 1509, 1022, 1750, 130

Starting from the current head position, what is the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests, for each of the following disk- scheduling algorithms?

- A. FCFS
- B. SSTF
- C. SCAN
- D. LOOK
- E. C-SCAN
- F. C-LOOK

Answer:-

A. The FCFS schedule is 143, 86, 1470, 913, 1774, 948, 1509, 1022, 1750, 130. The total seek distance is 7081.

B. The SSTF schedule is 143, 130, 86, 913, 948, 1022, 1470, 1509, 1750, 1774. The total seek distance is 1745.

C. The SCAN schedule is 143, 913, 948, 1022, 1470, 1509, 1750, 1774, 4999, 130, 86. The total seek distance is 9769.

D. The LOOK schedule is 143, 913, 948, 1022, 1470, 1509, 1750, 1774, 130, 86. The total seek distance is 3319.

E. The C-SCAN schedule is 143, 913, 948, 1022, 1470, 1509, 1750, 1774, 4999, 0, 86, 130. The total seek distance is 9985.

F. The C-LOOK schedule is 143, 913, 948, 1022, 1470, 1509, 1750, 1774, 86, 130. The total seek distance is 3363.

Paging

Answer:-

Paging is a memory-management scheme that permits the physical-address space of a process to be noncontiguous. Paging avoids the considerable problem of fitting memory chunks of varying sizes onto the backing store; most memory-management schemes used before the introduction of paging suffered from



(Subjective-CS604 OPERATING SYSTEM) **(From Final Term Papers “No repeated Questions”) June 2014**

this problem. The problem arises because, when some code fragments or data residing in main memory need to be swapped out, space must be found on the backing store. The backing store also has the fragmentation problems discussed in connection with main memory, except that access is much slower, so compaction is impossible. Because of its advantages over earlier methods, paging in its various forms is commonly used in most operating systems.

Characteristics of storage

Answer:-

Storage technologies at all levels of the storage hierarchy can be differentiated by evaluating certain core characteristics as well as measuring characteristics specific to a particular implementation. These core characteristics are volatility, mutability, accessibility, and addressability. For any particular implementation of any storage technology, the characteristics worth measuring are capacity and performance.

Volatility

Non-volatile memory

Will retain the stored information even if it is not constantly supplied with electric power. It is suitable for long-term storage of information.

Volatile memory

Requires constant power to maintain the stored information. The fastest memory technologies of today are volatile ones (not a universal rule). Since primary storage is required to be very fast, it predominantly uses volatile memory.

Dynamic random-access memory

A form of volatile memory which also requires the stored information to be periodically re-read and re-written, or refreshed, otherwise it would vanish.

Static random-access memory

A form of volatile memory similar to DRAM with the exception that it never needs to be refreshed as long as power is applied. (It loses its content if power is removed.)

Several file space allocation methods can be implemented in one Operating system how space allocation methods are implemented in Linux/ UNIX OS briefly explain.

Answer:-

Here are the three commonly used methods for file space allocation.



(Subjective-CS604 OPERATING SYSTEM)
(From Final Term Papers “No repeated Questions”) June 2014

Contiguous allocation

Linked allocation

Indexed allocation

UNIX Space Allocation

The UNIX file manager uses a combination of indexed allocation and linked lists for the index table. It maintains 10-15 direct pointers to file blocks, and three indirect pointers (one-level indirect, two-level indirect, and three-level indirect), all maintained in file's inode,

Banker's theorem ma sa 2 sa 3 question ayn ga so ap us ki calculation b kr lena.

MAX= allocation+ need

Available=sum of allocations- total etc