Advance Operating System CS 703 – Final Term Solutions

By: Imran Ullah Gondal(ms150200602)

Paper- 1 28-02-2016 10:30 AM

Q.2.Describe the main roles of a conflict resolution module?

Conflict Resolution
A mechanism that allows different device drivers to reserve hardware resources and to protect those resources from accidental use by another driver.
The conflict resolution module aims to:
- Prevent modules from clashing over access to hardware resources✓
- Prevent✓ *auto probes* from interfering with existing device drivers
- Resolve conflicts with multiple drivers trying to access the same hardware✓

Q.3.Consider a paging system with a page table stored in memory,

i) If a memory reference takes 200 nano seconds, how long does a paged reference takes (reference is retrieved from page table in memory and then program data is retrieved from memory) Assume all pages are in memory.

Answer: 400 nanoseconds. 200 ns to access the page table plus 200 ns to access the word in memory.

ii) If we add a TLB and 75% of all page table reference are found in TLB hit ratio is 75%,what effective memory reference time or average(Assume that finding a table entry in TLB takes 10 nanoseconds, if the entry is there)Assume all pages are in memory.

Answer: 75% * TLB hit-time + 25% * TLB miss-time = 75% * 200ns + 25% * 400ns = **250ns**

Q4.consider the following I/O scenarios on a single user PC
a). A disk drive containing user files.
b). A graphic card with direct bus connection, accessible through memory mapped I/O.

For each of these I/O scenarios would you design the OS to use buffering, Spooling, caching or a combination? Would you use parallel I/O or interrupt _driven I/O.
Answer:

a) A disk drive containing user files

Buffering can be used to hold data while in transit from user space to he disk, and versa. **Caching** can be used to hold disk-resident data for improved performance.

Spooling is not necessary because disks are shared-access devices. Interrupt driven I/O is best for devices such as disks that transfer data at slow rates.

b) A graphics card with direct bus connection, accessible through memory-mapped I/O
Buffering may be needed to control multiple access and for performance (double-buffering can be used to hold the next screen image while displaying the current one).
Caching and spooling are not necessary due to the fast and shared-access natures of the device. Polling and **interrupts** are only useful for input and for I/O completion detection, neither of which is needed for a memory-mapped device.

Q.5 Discretionary Access Control vs Mandatory Access Control
Discretionary Access Control
In scretionary access control (DAC), the owner of the object specifies which subjects can access the object. This model is called discretionary because the control of access is based on the discretion of the owner.
Most operating systems such as all Windows, Linux, and Macintosh and most flavors of Unix are based on DAC models.
In these operating systems, when you create a file, you decide what access privileges you want to give to other users; when they access your file, the operating system will make the access control decision based on the access privileges you created.

Mandatory Access Control
In mandatory access control (MAC), the system (and not the users) specifies which subjects can access specific data objects.

The MAC model is based on security labels. Subjects are given a security clearance (secret, top secret, confidential, etc.), and data objects are given a security classification (secret, top secret, confidential, etc.). The clearance and classification data are stored in the security labels, which are bound to the specific subjects and objects.

When the system is making an access control decision, it tries to match the clearance of the subject with the classification of the object. For example, if a user has a security clearance of secret, and he requests a data object with a security classification of top secret, then the user will be denied access because his clearance is lower than the classification of the object.

The MAC model is usually used in environments where confidentiality is of utmost importance, such a military institution. Examples of the MAC-based commercial systems are SE Linux and Trusted Solaris.

Q.1 Design principles of security

Solution:
• **Isolation**
✓ Separate processes execute in separate memory space
✓ Process can only manipulate allocated pages
• **Authentication**
✓ Who can access the system. Involves proving identities to the system
• **Access control**
✓ When can process create or access a file?
✓ Create or read/write to socket?
✓ Make a specific system call?
• **Protection problem**
✓ Ensure that each object is accessed correctly and only by those processes that are allowed to do so
• **Comparison between different operating systems**
✓ Compare protection models: which model supports least privilege most effectively?
✓ Which system best enforces its protection model?

Q.2 disk scheduling methods

FCFS (do nothing)
o Reasonable when load is low
o Long waiting times for long request queues
SSTF (shortest seek time first)
o Minimize arm movement (seek time), maximize request rate
o Favors middle blocks
SCAN (elevator)
o Service requests in one direction until done, then reverse
C-SCAN
o Like SCAN, but only go in one direction (typewriter)

Q.3 Some DMA controllers support direct virtual memory access, where the targets of I/O operations are specified as virtual addresses and a translation from virtual to physical address is performed during the DMA.

- How does this design complicate the design of the DMA controller?
- What are the advantages of providing such functionality?

Answer: Direct virtual memory access allows a device to perform a transfer from two memory-mapped devices without the intervention of the CPU or the use of main memory as a staging ground; the device simply issues memory operations to the memory-mapped addresses of a target device and the ensuing virtual address translation guarantees that the data is transferred to the appropriate device. This functionality, however, comes at the cost of having to support virtual address translation on addresses accessed by a DMA controller and requires the addition of an address-translation unit to the DMA controller. The address translation results in both hardware and software costs and might also result in coherence problems between the data structures maintained by the CPU for address translation and corresponding structures used by the DMA controller. These coherence issues would also need to be dealt with and results in further increase in system complexity.

Q.8 A table was given having segment number, base and size. For example:

Segment	base	size
0	3 4	34 3343a
1	3	343

To find out physical address for the following logical addresses for the above:

a) 3434
b) 34
C) 35

ANOLOGY

Q.9 :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

d) 1327+400=1727 e) illegal reference.

+1 for each correct solution.
For part c) and e), you get both points as long as that you recognize there are some sort of error.

Long Questions

Question 1:Write down detail at least five file operation and their corresponding UNIX/LINUX system calls. RR=4

Unix	NT
create(name)	CreateFile(name, CREATE)
open(name, mode)	CreateFile(name, OPEN)
read(fd, buf, len)	ReadFile(handle, …)
write(fd, buf, len)	WriteFile(handle, …)
sync(fd)	FlushFileBuffers(handle, …)
seek(fd, pos)	SetFilePointer(handle, …)
close(fd)	CloseHandle(handle, …)
unlink(name)	DeleteFile(name)
rename(old, new)	CopyFile(name)
	MoveFile(name)

Question 2: What are five major activities of an operating system in regard to file management?

Answer: just to remember name.

Operating systems perform the vital function of being the bridge between a computer's hardware and software. They provide an environment where software can be written without the need to cater to the specifics of the underlying hardware, which was necessary in the earlier days of computing. There are several widely-used operating systems which differ from each other in many respects. However, they each perform a number of similar functions including executing basic instructions, either compiled or interpreted; and also managing processes, memory, input and output, storage, network operations, and file and folder/directory operations.

There are five major file management functions that an operating system controls.

Creating and Deleting Files:

File creation and deletion are fundamental to computer operations. In the former, data can't be stored in an efficient manner unless arranged in some form of file structure. In the latter, permanent storage would quickly fill up if files were not deleted and the space occupied by them reallocated to new files.

Creating and Deleting Directories:

As a corollary to the need to store data in files, files themselves need to be arranged in directories or folders in order to allow their efficient storage and retrieval. This is particularly so in the case of personal computers where the user needs to navigate to one or more specific files to access them. Without some form of compartmentalization, this would prove an onerous if not impossible task. Much like file deletion, unnecessary directories or folders need to be removed in order to keep the system uncluttered.

File Manipulation Instructions:

Since operating systems allow application software to perform file manipulation using symbolic instructions, the operating system itself needs to have a machine-level instruction set in order to interface with the hardware directly. The application's symbolic instructions need to be translated into the machine-level instructions either by an interpreter or by compiling the application code. The operating system contains provisions to manage this machine-level file manipulation.

Mapping to Permanent Storage:

Operating systems need to be able to map files and folders to their physical location on permanent storage in order to be able to store and retrieve them. This will be recorded in some form of disk directory which varies according to the file system or systems that the operating system uses. The operating system will include a mechanism to locate the separate file segments where it has divided a file.

Backing Up Files:

Files represent a considerable investment in time, intellectual effort and often money as well, thus their loss can have a severe impact. Computer's permanent storage devices generally contain a number of mechanical devices which can fail, and the storage media itself may degrade. A function of operating systems is to obviate the risk of data loss by backing files up on additional secure and stable media in a redundant system.

Question3: How block devices are different from character devices? RR = 4

Answer:

- Character Device Vs. Block Device
  - A Character ('c') Device is one with which the Driver communicates by sending and receiving single characters (bytes, octets).
  - A Block ('b') Device is one with which the Driver communicates by sending entire blocks of data.
  - Examples for Character Devices: serial ports, parallel ports, sounds cards.
  - Examples for Block Devices: hard disks, USB cameras, Disk-On-Key.
  - For the user, the type of the Device (block or character) does not matter - you just care that this is a hard disk partition or a sound card.
  - Driver programmers, however, do care, but that's beyond our scope.

Block devices are those that can be read by blocks even though some programs may want to read char by char. These include tapes and hard drives.

Character devices are those only read char by char such as keyboards and serial ports.

A block device is one that is designed to operate in terms of the block I/O supported by Digital UNIX. It is accessed through the buffer cache. A block device has an associated block device driver that performs I/O by using file system block-sized buffers from a buffer cache supplied by the kernel. Block device drivers are particularly well-suited for disk drives, the most common block devices.

A character device is any device that can have streams of characters read from or written to it. A character device has a character device driver associated with it that can be used for a device such as a line printer that handles one character at a time. However, character drivers are not limited to performing I/O a single character at a time (despite the name ``character'' driver). For example, tape drivers frequently perform I/O in 10K chunks. A character device driver can also be used where it is necessary to copy data directly to or from a user process. Because of their flexibility in handling I/O, many drivers are character drivers. Line printers, interactive terminals, and graphics displays are examples of devices that require character device drivers.

Question 4: Consider the I/O scenario in single user PC.

a. A disk drive containing user files.
b. A graphics card with direct bus connection, accessible through memory mapped I/O.
c. A mouse used with a graphical user interface
d. A tape drive on a multitasking operating system (assume no device preallocation is available)

For each of these I/O scenario, would you design the operating system to use buffering, spooling, caching, or a combination? Would you use polled I/O, or interrupt driven I/O? Give reasons.

Answer:

a) A disk drive containing user files Buffering can be used to hold data while in transit from user space to the disk, and visa versa. Caching can be used to hold disk-resident data for improved perfor-mance. Spooling is not necessary because disks are shared-access devices. Interrupt-driven I/O is best for devices such as disks that transfer data at slow rates.

b) A graphics card with direct bus connection, accessible through memory-mapped I/O. Buffering may be needed to control multiple access and for performance (double-buffering can be used to hold the next screen image while displaying the current one).

Caching and spooling are not necessary due to the fast and shared-access natures of the device. Polling and interrupts are only useful for input and for I/O completion detection, neither of which is needed for a memory-mapped device.

c) A mouse used with a graphical user interface Buffering may be needed to record mouse movement during times when

higher-priority operations are taking place. Spooling and caching are inappropriate. Inter-rupt driven I/O is most appropriate.

d) A tape drive on a multitasking operating system (assume no device pre allocation is available). Buffering may be needed to manage throughput difference between the tape drive and the source or destination of the I/O, Caching can be used to hold copies of data that resides on the tape, for faster access. Spooling could be used to stage data to the device when multiple users desire to read from or write to it. Interrupt driven I/O is likely to allow the best performance.

Question 5: Consider a system that sport 5000 users, suppose that you want to allow 4990 of these users to be able to access one file.

a) How would you specify this protection scheme in UNIX?
You can add each of the 4,990 users to a group, and then use the allow access to the file by that group. Alternately, on some Unix systems, you can create an Access Control List (ACL) and assign access according to that list.

b) Could you suggest another scheme that can be used more affectively for this purpose then the scheme provided by UNIX?
In this case, it would be easier to be able to specify a group of users that can not access the file and allow any user not on the list to have access. Another alternative is to password the file and give the password to users that you deem need access to the file.

Another Answer:

a. There are two methods for achieving this:
i. Create an access control list with the names of all 4990 users.
ii. Put these 4990 users in one group and set the group access accordingly. This scheme cannot always be implemented since user groups are restricted by the system.
b. The universe access information applies to all users unless their name appears in the access-control list with different access permission. With this scheme you simply put the names of the remaining ten users in the access control list but with no access privileges allowed.

(**a).** A mouse used with a graphical user interface Buffering may be needed to record mouse movement during times when higher-priority operations are taking place. Spooling and caching are inappropriate. Interrupt driven I/O is most appropriate.
(b) A graphics card with direct bus connection, accessible through memory-mapped I/O Buffering may be needed to control multiple access and for performance (double-buffering can be used to hold the next screen image while displaying the current one). Caching and spooling are not necessary due to the fast and shared-access natures of the device. Polling and interrupts are only useful for input and for I/O completion detection, neither of which is needed for a memory-mapped device.

a.There are two methods for achieving this:
i. Create an access control list with the names of all 4990 users.
ii. Put these 4990 users in one group and set the group access accordingly. This scheme cannot always be implemented since user groups are restricted by the system.
b. The universe access information applies to all users unless their name appears in the access-control list with different access permission. With this scheme you simply put the names of the remaining ten users in the access control list but with no access privileges allowed.

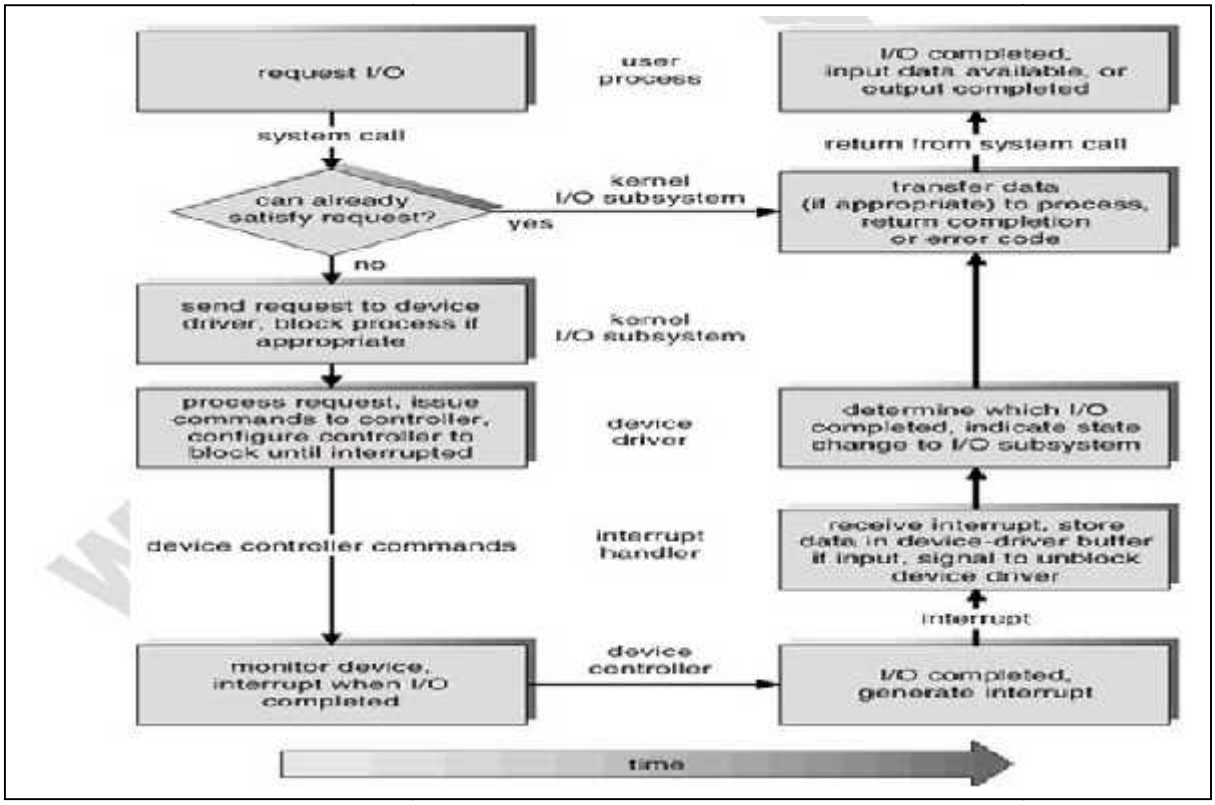
Q.6: Write the functionality of two lines put(blk, address)
get(address),,,,blk
Answer:

- put(blk, address) : writes data in blk on disk at address
- get(address) -> blk : returns blk at given disk address

Q.7: If there is no page frame free in page table. you want to page replacement. what step you taken for page replacement.
Answer:

- When there are no available free frames to handle a fault we must find a page to replace.
- How it does this is determined by the page replacement algorithm?
- The goal of the replacement algorithm is to reduce the fault rate by selecting the best victim page to remove.
- Write the victim frame to disk. Change all related page tables to indicate that this page is no longer in memory.

Question 8: Draw diagram of complete life cycle of an I/O request. RR = 3



Question 9:Calculate the affective access time in nanosecond and also drive a result that how much are the memory access during one page fault, if we take an average page fault service time of 30 millisecond and a memory access time of 100 nanoseconds. Also use the result and highlight whether it affects the performance or not. RR = 4

Answer:

Effective access time = (1-p)×(memory access time) + p×(page-fault service time)
=(1-p)*(100)+p*(30ms)
=(1-p)*100+p*(30000000ns)
=100-p100+p*30000000
=100+p(29000000)

Ans.

Effective access time $= (1-p) * (200) + p\ (30\ \text{milliseconds})$
$= (1-p) * 200 + p * 30{,}000{,}000$
$= 100 + 24{,}999{,}900 * p$

We see then that the effective access time is directly proportional to the page fault rate. If one access out of 1,000 causes a page fault, the effective access time is 25 microseconds. The computer would be slowed down by a factor of 250 because of demand paging! If we want less than 10 percent degradation, we need:

$110 > 100 + 25{,}000{,}000 * p$
$10 > 25{,}000{,}000 * p$
$p < 0.0000004$

That is, to keep the slowdown due to paging to a reasonable level, we can allow only less than one memory access out of 2,500,000 to page fault.

It is important to keep the slowdown due to paging to a reasonable level, we can allow only less than one memory access out of 2,500,000 to page fault.

It is important to keep the page fault rate low in a demand-paging system. Otherwise the effective access time increases, slowing process execution dramatically.

Question 10: **Why Rotational latency is not considered in disk scheduling ? How FCFS, SSTF, SCAN and C-SCAN are modified for rotational latency optimization RR=3**

Answer:

Most disks do not export their rotational position information to the host. Even if they did, the time for this information to reach the scheduler would be subject to imprecision and the time consumed by the scheduler is variable, so the rotational position information would become incorrect. Further, the disk requests are usually given in terms of logical block numbers, and the mapping between logical blocks and physical locations is very complex.

Question 11: **In Unix file system demand paging transfer time 20 millisecond and access time..... (Non remembered)**
Answer:

EAT = (0.8 * 1 us) + (0.18 * 2 us) + (0.02 * (20000us + 2us)) = 0.8 us + 0.36 us + 400.04 us = 401.2 us

Question 12: i Node and pointer 10K..... (Non remembered) RR = 2

Answer:

File sharing is the public or private sharing of computer data or space in a network with various levels of access privilege. While files can easily be shared outside a network (for example, simply by handing or mailing someone your file on a diskette), the term file sharing almost always means sharing files in a network, even if in a small local area network. File sharing allows a number of people to use the same file or file by some combination of being able to read or view it, write to or modify it, copy it, or print it. Typically, a file sharing system has one or more administrators. Users may all have the same or may have different levels of access privilege. File sharing can also mean having an allocated amount of personal file storage in a common file system.

More usually, however, file sharing implies a system in which users write to as well as read files or in which users are allotted some amount of space for personal files on a common server, giving access to other users as they see fit. The latter kind of file sharing is common in schools and universities. File sharing can be viewed as part of file systems and their management. Any multi-user operating system will provide some form of file sharing. Among the best known network file systems is (not surprisingly) the Network File System (NFS). it lets you read and, assuming you have permission, write to sharable files as though they were on your own personal computer. Files can also be shared in file systems distributed over different points in a network. File sharing is involved in groupware and a number of other types of applications.

Question 13: Three blocking I/O and three nonblocking I/O. Why nonblocking I/O is useful for busy wait 10 marks

Answer: Generally, blocking I/O is appropriate when the process will only be waiting for one specific event. Examples include a disk, tape, or keyboard read by an application program. Non-blocking I/O is useful when I/O may come from more than one source and the order of the I/O arrival is not predetermined. Examples include network daemons listening to more than one network socket, window managers that accept mouse movement as well as keyboard input, and I/O-management programs, such as a copy command that copies data between I/O devices. In the last case, the program could optimize its performance by buffering the input and output and using nonblocking I/O to keep both devices fully occupied. Non-blocking I/O is more complicated for programmers, because of the asynchronous rendezvous that is needed when an I/O occurs. Also, busy waiting is less efficient than interrupt-driven I/O so the overall system performance would decrease

Blocking - process suspended until I/O completed ⎪ Easy to use and understand ⎪ Insufficient for some needs

Nonblocking - I/O call returns as much as available ⎪ User interface, data copy (buffered I/O) ⎪ Implemented via multi-threading inside the kernel ⎪ Returns quickly with count of bytes read or written

Question 14: Describe in details file sharing system.
Answer:

File sharing is the practice of distributing or providing access to digital media, such as computer programs, multimedia(audio, images and video), documents or electronic books. File sharing may be achieved in a number of ways. Common methods of storage, transmission and dispersion include manual sharing utilizing removable media, centralized servers on computer networks, World Wide Web-based hyperlinked documents, and the use of distributed peer-to-peer networking.

Peer-to-peer file sharing

Peer-to-peer file sharing is based on the peer-to-peer (P2P) application architecture. shared files on the computers of other users are indexed on directory servers. P2P technology was used by popular services like Napster, Spotify, and Infinit. The most popular protocol for P2P sharing is BitTorrent.

File sync and sharing services

Cloud-based file syncing and sharing services implement automated file transfers by updating files from a dedicated sharing directory on each user's networked devices. Files placed in this folder also are typically accessible through a website and mobile app, and can be easily shared with other users for viewing or collaboration. Such services have become popular via consumer products such as Dropbox and Google Drive.

Q.15: When a program enters into an infinite loop and never returns control back to CPU. Explain how timer interrupt help to terminate this condition RR=3

Answer: An infinite loop (also known as an endless loop or unproductive loop) is a sequence of instructions in a computer program which loops endlessly, either due to the loop having no terminating condition, having one that can never be met, or one that causes the loop to start over. In older operating systems with cooperative multitasking, infinite loops normally caused the entire system to become unresponsive. With the now-prevalent preemptive multitasking model, infinite loops usually cause the program to consume all available processor time, but can usually be terminated by the user. Busy wait loops are also sometimes called "infinite loops". One possible cause of a computer "freezing" is an infinite loop; others include thrashing, deadlock, and access violations.

Q.16 if there is no page frame free in page table. you want to page replacement. what step you taken for page replacement.

Answer:

When a process references a virtual address in a page that has been evicted then in this case a page fault occurs. So, in this case,

- when the page was evicted, the OS set the PTE as invalid and noted the disk location of the page in a data structure (that looks like a page table but holds disk addresses)
- when a process tries to access the page, the invalid PTE will cause an exception (page fault) to be thrown. This page fault is known as a kind of interupt.
- the OS will run the page fault handler in response
  - handler uses the “like a page table” data structure to locate the page on disk
  - handler reads page into a physical frame, updates PTE to point to it and to be valid
  - OS restarts the faulting process.

Now the matter of page replacement frame can be handled in a way as:

- When you read in a page, where does it go?
  - if there are free page frames, grab one
    - what data structure might support this?
  - if not, must evict something else
  - this is called page replacement
- Page replacement algorithms
  - try to pick a page that won’t be needed in the near future
  - try to pick a page that hasn’t been modified (thus saving the disk write)
  - OS typically tries to keep a pool of free pages around so that allocations don’t inevitably cause evictions
  - OS also typically tries to keep some “clean” pages around, so that even if you have to evict a page, you won’t have to write it
    - accomplished by pre-writing when there’s nothing better to do

Q.19. Consider a virtual memory in which a process does not have enough pages therefore the page fault rate is very high. This leads to low CPU and I/O devices utilization, where as disk utlization is high. Write down the name of the activity that leads to said problems and how can you overcome these problems.

Answer: Thrashing is occur.

Q.20: what is a working set model ?what are the difficulties of it and also the solution.

Answer: The **working set model:** The working set of a process is used to model the dynamic locality of its memory usage ⏐ working set = set of pages process currently "needs" ⏐ formally defined by Peter Denning in the 1960‟s • Definition: ⏐ a page is in the working set (WS) only if it was referenced in the last w references ⏐ obviously the working set (the particular pages) varies over the life of the program ⏐ so does the working set size (the number of pages in the WS)

Some problems • T is magic ⏐ what if T too small? Too large? ⏐ How did we pick it? Usually "try and see" ⏐ Fortunately, system‟s aren‟t too sensitive • What processes should be in the balance set? ⏐ Large ones so that they exit faster? ⏐ Small ones since more can run at once? • How do we compute working set for shared pages.

Q#21:I-node Area.

Answer: -i-node area ⏐ contains descriptors (i-nodes) for each file on the disk; all i-nodes are the same size; head of freelist is in the superblock. Each i-node contains 13 block pointers • first 10 are "direct pointers" (pointers to 512B blocks of file data) • then, single, double, and triple indirect pointers

Q#22. Design:Design Principles for Security

1. System design should be public
2. Default should be no access
3. Check for current authority
4. Give each process least privilege possible
5. Protection mechanism should be - simple - uniform - in lowest layers of system
6. Scheme should be psychologically acceptable

Biometrics • Biometrics ⏐ Authentication of a person based on a physiological or behavioral characteristic
Example features: ⏐ Face, Fingerprints, Hand geometry, Handwriting, ⏐ Iris, Retinal, Vein, Voice. • Strong authentication but still need a "Trusted Path".

Q#23. Briefly explain major security issues. RR = 3

Answer:

- **Isolation**
  - Separate processes execute in separate memory space Process can only manipulate allocated pages
- **Authentication**
  - Who can access the system. Involves proving identities to the system
- **Access control**
  - When can process create or access a file? Create or read/write to socket?
  - Make a specific system call?
- **Protection problem**
  - Ensure that each object is accessed correctly and only by those processes that are allowed to do so
- **Comparison between different operating systems**
  - Compare protection models: which model supports least privilege most effectively?
  - Which system best enforces its protection model?

Question 24: How authentication can be achieve by biometric?

Answer:

Authentication of a person based on a physiological or behavioral characteristic. A number of biometric methods have been introduced over the years, but few have gained wide acceptance.

Signature dynamics: Based on an individual's signature, but considered unforgeable because what is recorded isn't the final image but how it is produced -- i.e., differences in pressure and writing speed at various points in the signature.

Typing patterns: Similar to signature dynamics but extended to the keyboard, recognizing not just a password that is typed in but the intervals between characters and the overall speeds and pattern. This is akin to the way World War II intelligence analysts could recognize a specific covert agent's radio transmissions by his "hand" -- the way he used the telegraph key.

Eye scans: This favorite of spy movies and novels presents its own problems. The hardware is expensive and specialized, and using it is slow and inconvenient and may make users uneasy.
In fact, two parts of the eye can be scanned, using different technologies: the retina and the iris.

Fingerprint recognition: Everyone knows fingerprints are unique. They are also readily accessible and require little physical space either for the reading hardware or the stored data.

Hand or palm geometry: We're used to fingerprints but seldom think of an entire hand as an individual identifier. This method relies on devices that measure the length and angles of individual fingers. Although more user-friendly than retinal scans, it's still cumbersome.

Voice recognition: This is different from speech recognition. The idea is to verify the individual speaker against a stored voice pattern, not to understand what is being said.

Facial recognition: Uses distinctive facial features, including upper outlines of eye sockets, areas around cheekbones, the sides of the mouth and the location of the nose and eyes. Most technologies avoid areas of the face near the hairline so that hairstyle changes won't affect recognition.

Q # 26. What are five major activities of an operating system in regard to file management?

Answer: The five main major activities of an operating system in regard to file management are

1. The creation and deletion of files.
2. The creation and deletion of directions.
3. The support of primitives for manipulating files and directions.
4. The mapping of files onto secondary storage.
5. The back up of files on stable storage media.