Past Exams

> @author Suleyman Golbol

2016 - Final

General OS

a) What are the three main services that an OS provide?

1. Managing the computer's resources, such as the CPU, RAM(main memory), disk drives, and printers
2. Establishing an user interface
3. Executing and providing services for applications software
4. Being an abstraction level between hardware and the user

b) What do we mean “Ontogeny Recapitulates Phylogeny”. Give an example.

- Each new generation of computer systems goes through the same phases of its ancestor.
- Mainframes-mini computer - … - smart card

c) What is POSIX?

- POSIX is a family of standards, specified by the IEEE to clarify and make uniform the application programming interfaces provided by UNIX operating systems
- When we write our programs to rely on POSIX standards, we can be pretty sure to be able to port them easily among a large family of Unix derivatives
- Portable Operating System Interface for UNIX

Processes/Threads

a) **Explain thread_yield() function. Is there process_yield() function? If no, why?**

- Threads cooperates unlike process and they know there are other threads other than themselves. So, they can let the other threads to use CPU and other resources with parallelism idea via using thread_yield() function.
- There are no process_yield() because all processes thinks they have their own CPU and other recurces, they don't know another process running and using these resources at the same time. Processes don't cooperate.

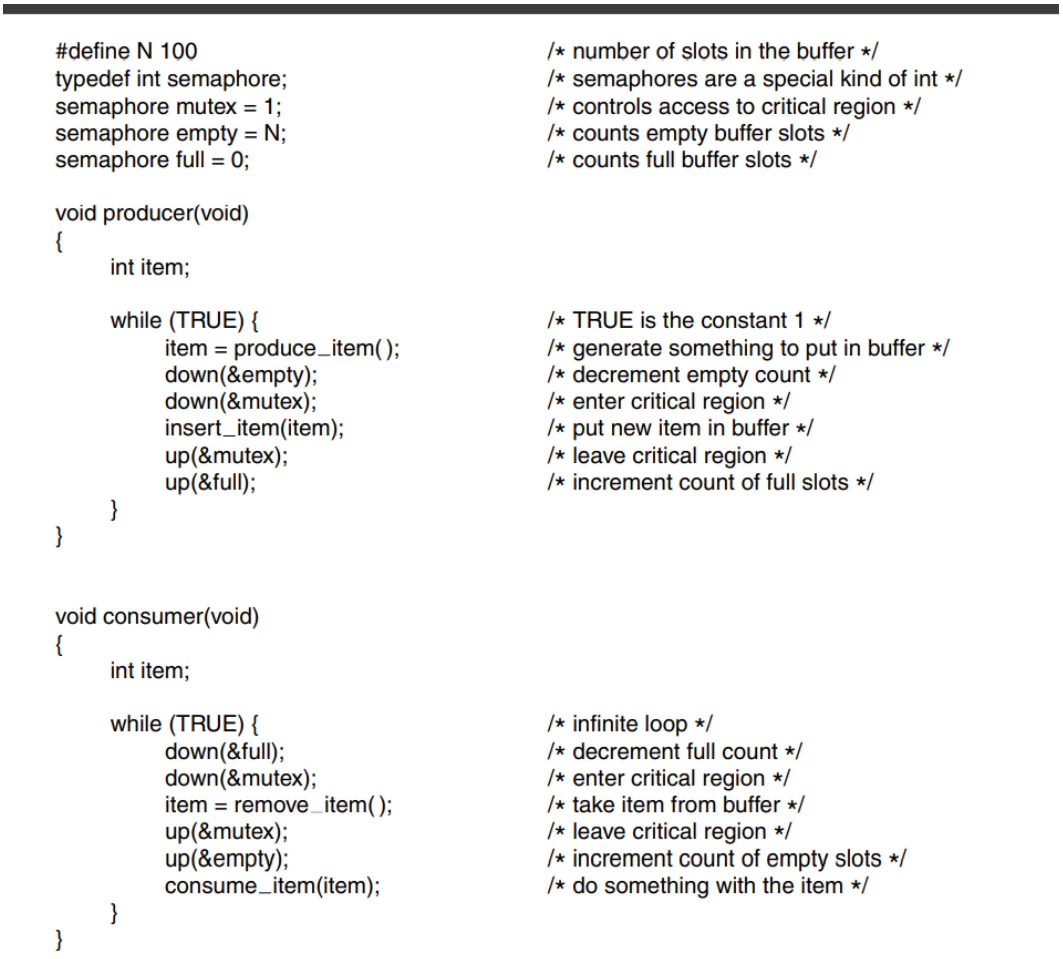
b) **Explain how time quantum value and context switching time affect each other, in a round robin scheduling alogrithm.**

- When the time quantum is very large, the Round Robin policy becomes a FCFS(first come first server) policy. It will cause poor response time to short interactive processes

Explanation: The period of time for which a process is allowed to run in a pre-emptive multitasking system is generally called the time slice or quantum. Each process is assigned a fixed time (Time Quantum/Time Slice) in cyclic way.

- When the time quantum is too short, it causes too many process/context switches and reduces CPU efficiency.

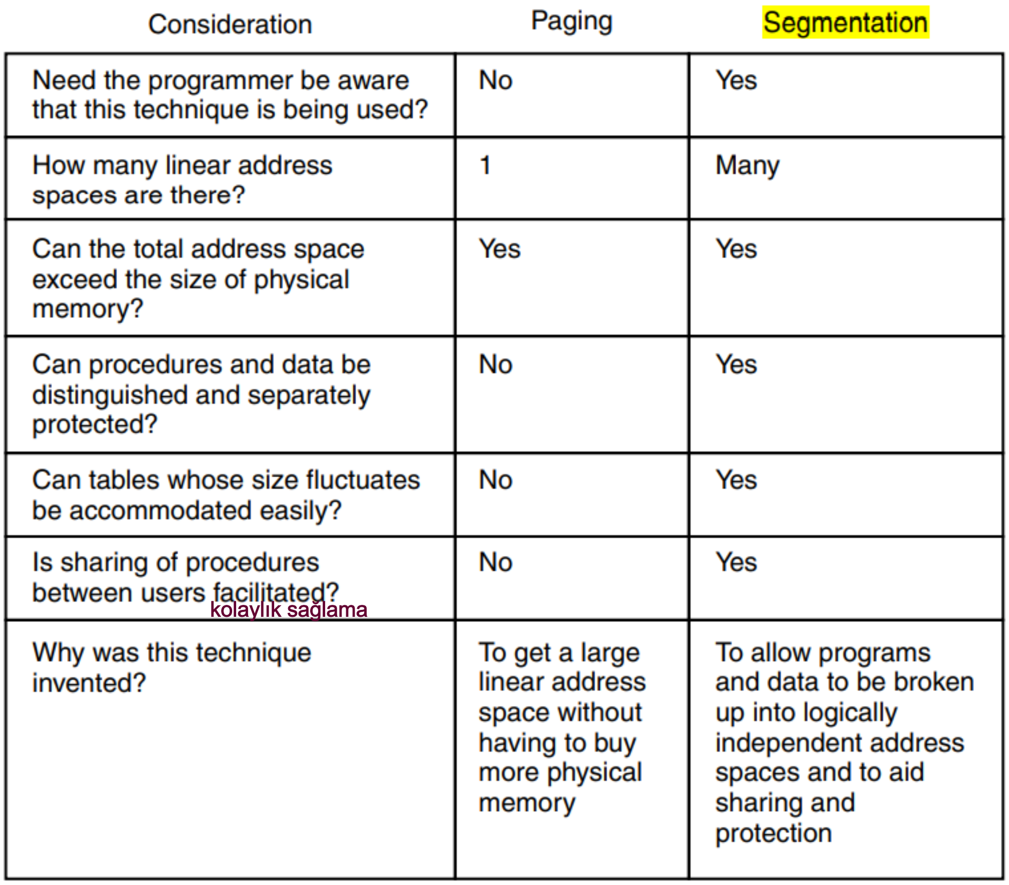
c) **Implement 1 producer and 2 consumer processes using semaphore.**



```
#define N 100                           /* number of slots in the buffer */
typedef int semaphore;                  /* semaphores are a special kind of int */
semaphore mutex = 1;                    /* controls access to critical region */
semaphore empty = N;                    /* counts empty buffer slots */
semaphore full = 0;                     /* counts full buffer slots */

void producer(void)
{
        int item;

        while (TRUE) {                  /* TRUE is the constant 1 */
                item = produce_item( );  /* generate something to put in buffer */
                down(&empty);           /* decrement empty count */
                down(&mutex);           /* enter critical region */
                insert_item(item);      /* put new item in buffer */
                up(&mutex);             /* leave critical region */
                up(&full);              /* increment count of full slots */
        }
}


void consumer(void)
{
        int item;

        while (TRUE) {                  /* infinite loop */
                down(&full);            /* decrement full count */
                down(&mutex);           /* enter critical region */
                item = remove_item( );  /* take item from buffer */
                up(&mutex);             /* leave critical region */
                up(&empty);             /* increment count of empty slots */
                consume_item(item);     /* do something with the item */
        }
}
```

Memory

Segmentation in OS divides the secondary memory into uneven-sized blocks known as segments. It divides processes into smaller sub-parts known as modules. Segmentation is done considering that the relative data should come in a single segment. Partition size depends upon the type and length of modules.



Consideration	Paging	Segmentation
Need the programmer be aware that this technique is being used?	No	Yes
How many linear address spaces are there?	1	Many
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? kolaylık saglama	No	Yes
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

Figure 3-32. Comparison of paging and segmentation.

a) What are the advantages of using segmentation with paging instead of using only paging?

- When a segment of the memory grows and shrinks, it affects other segments (start-end points can change, etc.) Segmentation allows these segments to be in separate virtual address spaces so they don't affect each other.

b) What is an inverted page table?

- Inverted Page Table is the global page table which is maintained by the Operating System for all the processes.
- In inverted page table, the number of entries is equal to the number of frames in the main memory.
- It can be used to overcome the drawbacks of page table.

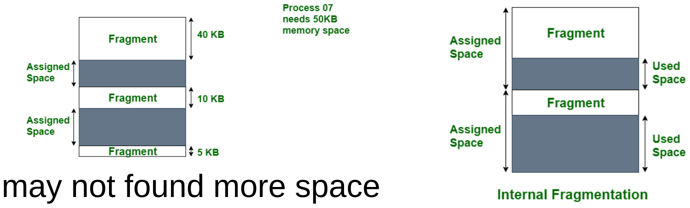
File System

a) How does the block cache reduce the disc access?

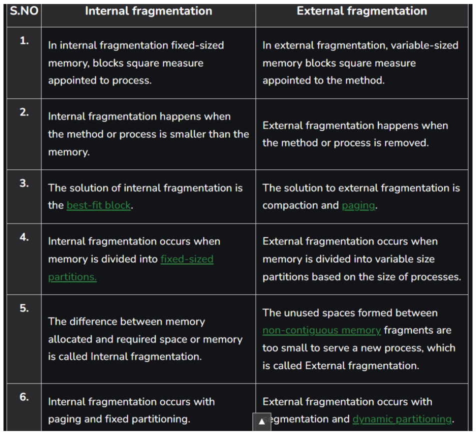
- A block cache is a collection of blocks that logically belong on the disk but are being kept in memory. So, these blocks can be accessed without going to the disk

b) What are the advantages and disadvantages of contiguous and table file allocation?

- **Contiguous file allocation advantages:**
  - Easy to implement (only 2 numbers are required. Starting address and number of blocks)
  - Read performance is excellent because the entire file can be read from the disk in a single operation
  - Faster because there is no deletion or modification allowed (only used for cd roms)
- **Contiguous file allocation disadvantages:**
  - External Fragmentation
  - (sanırım) If we need more space then we may not found more space contiguously, since we kept just 2 numbers we cant kept the other addresses belongs to that file.



- **List file allocation advantages:**
  - No external fragmentation.
- **List file allocation disadvantages:**
  - Entire table must be in memory all the time.



S.NO	Internal fragmentation	External fragmentation
1.	In internal fragmentation fixed-sized memory, blocks square measure appointed to process.	In external fragmentation, variable-sized memory blocks square measure appointed to the method.
2.	Internal fragmentation happens when the method or process is smaller than the memory.	External fragmentation happens when the method or process is removed.
3.	The solution of internal fragmentation is the best-fit block.	The solution to external fragmentation is compaction and paging.
4.	Internal fragmentation occurs when memory is divided into fixed-sized partitions.	External fragmentation occurs when memory is divided into variable size partitions based on the size of processes.
5.	The difference between memory allocated and required space or memory is called Internal fragmentation.	The unused spaces formed between non-contiguous memory fragments are too small to serve a new process, which is called External fragmentation.
6.	Internal fragmentation occurs with paging and fixed partitioning.	External fragmentation occurs with segmentation and dynamic partitioning.

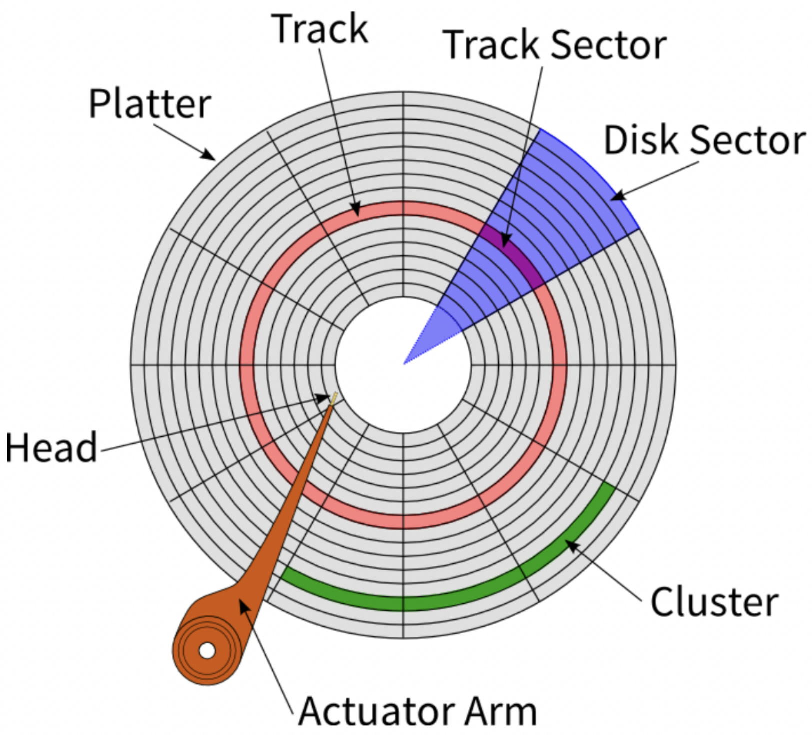
Input/Output

a) What is the difference between I/O and DMA?

- DMA allows the disk controllers access directly to the memory. In that way devices can directly write/read into memory.
- [CBB] In I/O we should use OS to get datas from the device controllers. That means more interrupt will happen and more CPU time will be wasted.

b) Define disc tracks, sectors, zones, cylinders? Draw their figures.

- **Disk track:** It is a circular path on the surface of a disk on which information is magnetically recorded and from which recorded information is read
- **Disk Sector:** The sector is the minimum storage unit of a hard drive. In computer ***disk*** storage, a sector is a subdivision of a ***track*** on a magnetic ***disk.*** *Each sector stores a fixed amount of user-accessible data.*
- **Cylinders:** A cylinder is a division of data in a disk drive



2017 - Final

General OS

a) What are the differences between system calls and procedure calls?

- System calls are calls from user space to kernel space. You are asking the OS to do something for you.
- Procedure calls are calls from one user space program to another. You are asking another process to do something for you.

- A system call is a primitive function in the system kernel which can be called from applications. It’s the lowest level of abstraction into the OS, and the only low-level entry points available to applications.

b) **What is the BIOS and purpose of bios?**

- BIOS, basic input/output system, is a program which is stored in non-volatile memory(ROM).
- Main function of BIOS is to set up hardware and start an OS.
- Also it examines all the hardware connections and detects all of the devices.
- First level of OS abstraction starts with BIOS.

c) **What do we mean by “Ontogeny Recapitulates Phylogeny”**

- Each new generation of computer systems goes through the same phases of its ancestor.
- Mainframes-mini computer - … - smart card.

Processes/Threads

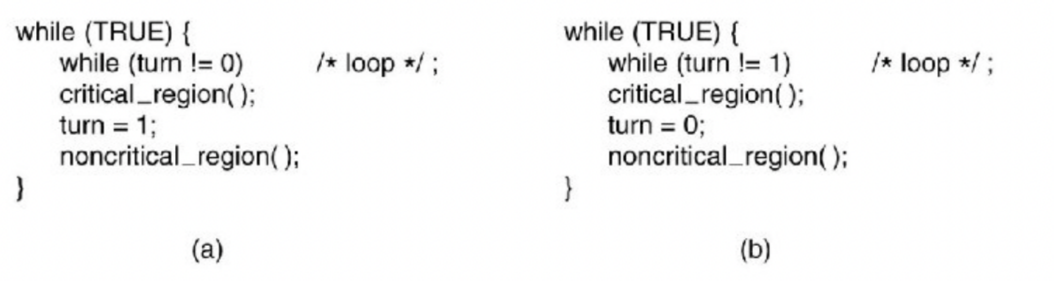
a) **Can we sent an interrupt to a thread?**

- If the thread is implemented at kernel space, the OS can send a timer interrupt to the thread. Also, another thread can send an interrupt to the thread.

b) **Strict alternation for N process? (soru ne anlamadım)**

- Not possible for more than 2 processes. For 2 processes it is like this:

its actally possible for n processes.



```
while (TRUE) {
     while (turn != 0)          /* loop */ ;
     critical_region( );
     turn = 1;
     noncritical_region( );
}
```

(a)

```
while (TRUE) {
     while (turn != 1)          /* loop */ ;
     critical_region( );
     turn = 0;
     noncritical_region( );
}
```

(b)

Memory (burada ne tarz sorular geldiğini söylemiş)

a) There is a code and TLB, Page size was given. Find values which not defined in code.

- Answer

b) Virtual Address translation to Physical Address and find page table entry size

- Answer

c) Internal and External fragmentation and their usage area, pure segmentation, paging

- Internal fragmentation: It occurs on the allocation of a process to a partition greater than the process's requirement.
  The leftover space causes degradation system performance

- External fragmentation happens when the method or process is removed.
  The solution to external fragmentation is compaction and paging.
  External fragmentation occurs when memory is divided into variable size partitions based
  on the size of processes.

File System

a) How can we find missed block in file system? (Soru enteresan)

- Tek tek bakarak, ya da aradığımız block yoksa yoktur diyerek 😵

**b) Address size, block size was given. There was a picture. Find the max file size in
system.**

- Answer

c) What is the direct memory write? What are the advantages of direct write?

- Aşağıda DMA(direct memory access) açıklanmış.

Input/Output

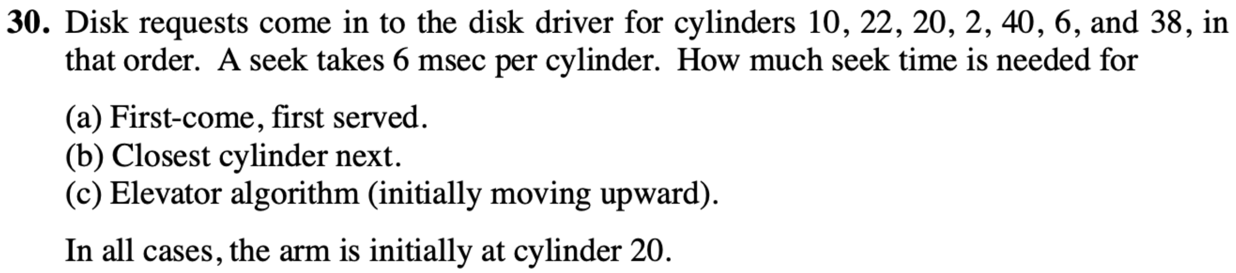
a) What is the DMA(Direct Memory Access)? In which status you don't need to use dma although you have a dma?

- Slow devices like keyboards will generate an interrupt to the main CPU after each byte is transferred. If a fast device such as a disk generates an interrupt for each byte, the operating system would spend most of its time handling these interrupts. So a typical computer uses direct memory access (DMA) hardware to reduce this overhead.
- DMA module itself controls exchange of data between main memory and the I/O device. CPU is only involved at the beginning and end of the transfer and interrupted only after entire block has been transferred.
- Thus, DMA is not needed if there are no fast devices on our computer such as disks(not possible without it but let's assume). If we only use I/O devices like keyboards which don't generate interrupts so often, then wo don't need to use DMA.

b) **What is double buffering?**

- Putting a secondary buffer to kernel space to avoid losing incoming data on writing the data on kernel space buffer to user space buffer.

c) **Block coming order was given. Like 5-26-18-41... Apply elavator, fifo and shortest seek time first, show steps and calculate seek times. (Muhtemelen aşağıdaki sorudan)**



30. Disk requests come in to the disk driver for cylinders 10, 22, 20, 2, 40, 6, and 38, in that order. A seek takes 6 msec per cylinder. How much seek time is needed for

(a) First-come, first served.
(b) Closest cylinder next.
(c) Elevator algorithm (initially moving upward).

In all cases, the arm is initially at cylinder 20.

- a) 10-22-20-2-40-6-38 (total seek time: 10*seektime + 12*seektime + 2*seektime + 18*seektime + 38*seektime + 34*seektime + 32*seektime = 146*seektime)
- b) 20-22-10-6-2-38-40 (total seek time: 0*seektime + 2*seektime + 12*seektime + 4*seektime + 4*seektime + 36*seektime + 2*seektime = 60*seektime)
- c) 20-22-38-40-10-6-2 (total seek time: 0*seektime + 2*seektime + 16*seektime + 2*seektime + 30*seektime + 4*seektime + 2*seektime = 56*seektime)

2018 - Final

(burdaki soruları öğrenciler yazmış o yüzden sorular yok kelimeler var ben önlerine what is gibi ekledim sadece)

General OS

a) What is hyperthreading?

- Rapidly switching between threads to create a perception of parallelism using preemption

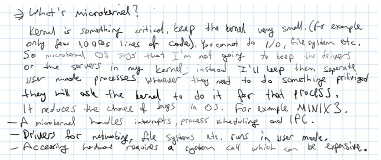
kesinti, öncelik değiştirme

b) What is Ontogeny Recapitulates Phylogeny?

- Each new generation of computer systems goes through the same phases of its ancestor.
- Mainframes-mini computer - … - smart card

c) What is a microkernel?

- A microkernel is a kernel implementation that keeps only the required OS stuff in the kernel and puts the rest in the user space. This makes the kernel more error-proof since one error in the kernel space can crash the whole system.



⇒ What's microkernel?
Kernel is something critical, keep the kernel very small. (for example only few 1000s lines of code). You cannot do I/O, file system etc. So microkernel OS says that I'm not going to keep the drivers or the servers in my kernel; instead I'll keep them seperate user mode processes. Whenever they need to do something prilivged they will ask the kernel to do it for that process.
It reduces the chance of bugs in OS. For example MINIX3.
– A microkernel handles interrupts, process scheduling and IPC.
– Drivers for networking, file systems etc. runs in user mode.
– Accessing hardware requires a system call which can be expensive.

d) What is preemption?

- Giving each process a quantum of time to run and switching between processes when they use their quantum. It is essential for interaction & multiprocessing.

Processes/Threads

a) Informations that are unique for each thread? (stack farklı mesela, address spaceler aynı)

- Program counter, registers, stack

b) How to solve synchronisation problem of 1 Producer - 2 Consumer Problem?

- Mutex, semaphore, condition variable falan

c) Explain roundrobin scheduling.

- Processes are kept in a circular linked list and scheduled in this order.

Memory

a) Why LRU is expensive?

- Because, it requires special hardware (64-bit counter which is incremented after each instruction). we have to keep track of which pages were referenced all the time.
- (İlk cevap şu şekilde geliştireilebilinir): There is two type of LRU, Hardware asistant and software asistant.
    - Hardware asistant is expensive since it requires an hardware.
    - Software asistant is also has downside since it slower than the hardware asistant one.

b) Differences between Two-Level Paging and Multi-Level Paging?

- Answer

But if the size of the page table is larger than the size of the Frame. Then the page table in return is to be divided into several pages and these pages of the page table are to be stored in the main memory. Thus, an Outer Page Table comes into the picture. This Outer Page Table would contain the address of the Frames which contain the pages of the Inner Page Table (i.e., Page Table one page) in the main memory. The size of this Outer Page is also calculated in the same way as explained above and was used to calculate the size of the inner page Table.

c) Can a page be in two working sets at the same time? Explain.

Working set is a concept in computer science which defines the amount of memory that a process requires in a given time interval.

- (Hiç emin değilim)Yes, there are shared pages that are used by multiple processes at the same time (e.g DLL's) (thanks to SHM)
- (Ben bi tık daha fazla eminim): Yes, like C librares, there are shared page and can be used in more than one procedure since they will not be modified.

File System

a) What are the differences between soft link and hard link?

- Hard link creates 2 different files which points to same location at the memory
- Soft link is just a pointer to the original file so if the original one is deleted, soft link doesn't work anymore. (Like windows, creating shortcut with right click)

Input/Output

a) What is RAID? Design a level of RAID that are good for disk read speed and another level that are good for disk crashes. Show your solutions with figures.

The idea behind the RAID is to keep data separated on different Disks and reading them in parallel to speed up disk reads.

- RAID (redundant array of independent disks) is a way of storing the same data in different places on multiple hard disks or SSD's to protect data in the case of a drive failure. There are different RAID levels, however, and not all have the goal of providing redundancy.yedeklilik

2018 - Tek Ders Sınavı

General OS

a) What is multiprogramming and timesharing?

- **Multiprogramming,** is an ability of an operating system that executes more than one program using a single processor machine.
- More than one task or program or jobs are present inside the main memory at one point of time.
- **Time sharing,** is a logical extension of multiprogramming.
- (Atkan Altından geldi): The main difference between multiprogramming and time-sharing is that multiprogramming effectively utilizes CPU time by allowing several programs to use the CPU simultaneously. But time-sharing is sharing a computing facility by several users who want to use the same facility simultaneously

b) What is MMU(memory management unit) and what it is used for?

- A part of the CPU
- A memory management unit (MMU) is **a computer hardware component that handles all memory and caching operations associated with the processor**. In other words, the MMU is responsible for all aspects of memory management.

c) We know context switching is expensive. In which time period context switch was more expensive (now or 1970's)?

- At 1970's, context swithcing was more expensive since, data transfer speed is more slower (cause we have improved cache system that increasese the speed), context switch was more uneffici

d) What is hardware protection?

- Hardware security is vulnerability protection that comes in the form of a physical device rather than software that is installed on the hardware of a computer system.

ilgili olmak , uygun olmak

Hardware security can pertain to a device used to scan a system or monitor network traffic. Common examples include hardware firewalls and proxy servers.

Processes/Threads

a) There are multiple programs which will be executed parallelly. Which one would you choose? Multithreading or multiprogramming? Why?

- (sanırım) I would choose multiprogramming since we have different programs we need different address spaces and protection. No other programs should access other programs address space. In multithreading they can access their address spaces and we don't want it for sake of protection.

if the programs need to share resources, communicate extensively, or require high coordination, multithreading can be a suitable choice. On the other hand, if the programs need strong isolation, fault tolerance, or security, multiprogramming can be more appropriate.

b) Assume we are using round robin scheduler on single CPU system. The context switch time is 5 msecs. We have 10 interactve processes and 10 batch process running on this system. We want to our interactive processes to respond in at least 200 msecs for user requests. What would be the quantum time for the interactive processes and batch processes? What would be the system efficiency?

- Interactive process should respons = 200msec.
  X= interactive, Y= batch to be fair between 10X = Y. System time = (200-55) / 200
  10X + Y + (11 * 5) < 200
  2Y < 145 ➔ Y < 77 msec.
  X < 145 / 20 → 7 msec.
- Ne saçma cevap

c) Implement a semaphore using Assembly STL

- Answer

Memory

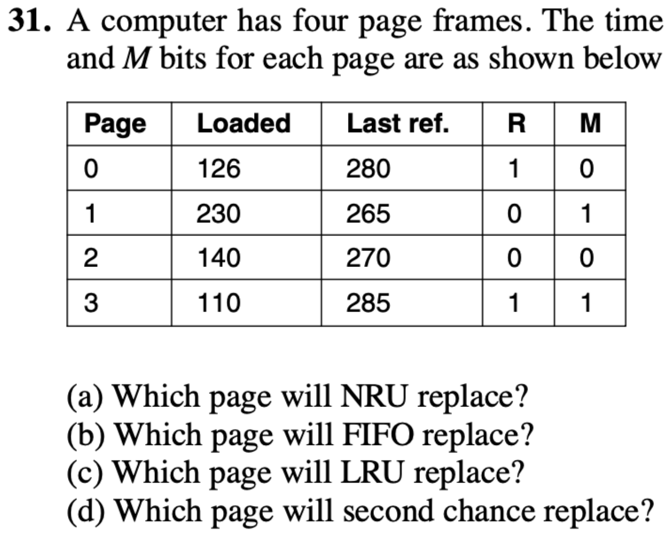
a) How to implement optimal page replacement algorithm?

- Optimal page replacement algorithm is not implementable since it needs to know how long each process will run in advance. It might be approximately implemented on batch systems since we approximately know how long each proccess will take. This algorithm usually ran on simulations to compare implementable algorithms with it.

b) There are 2^n bit virtual adress, 2^m bit pyhsical address and 2^L KB page size. What is the page size with single level paging?

- Answer

c) Bu soru hep çıkmış:



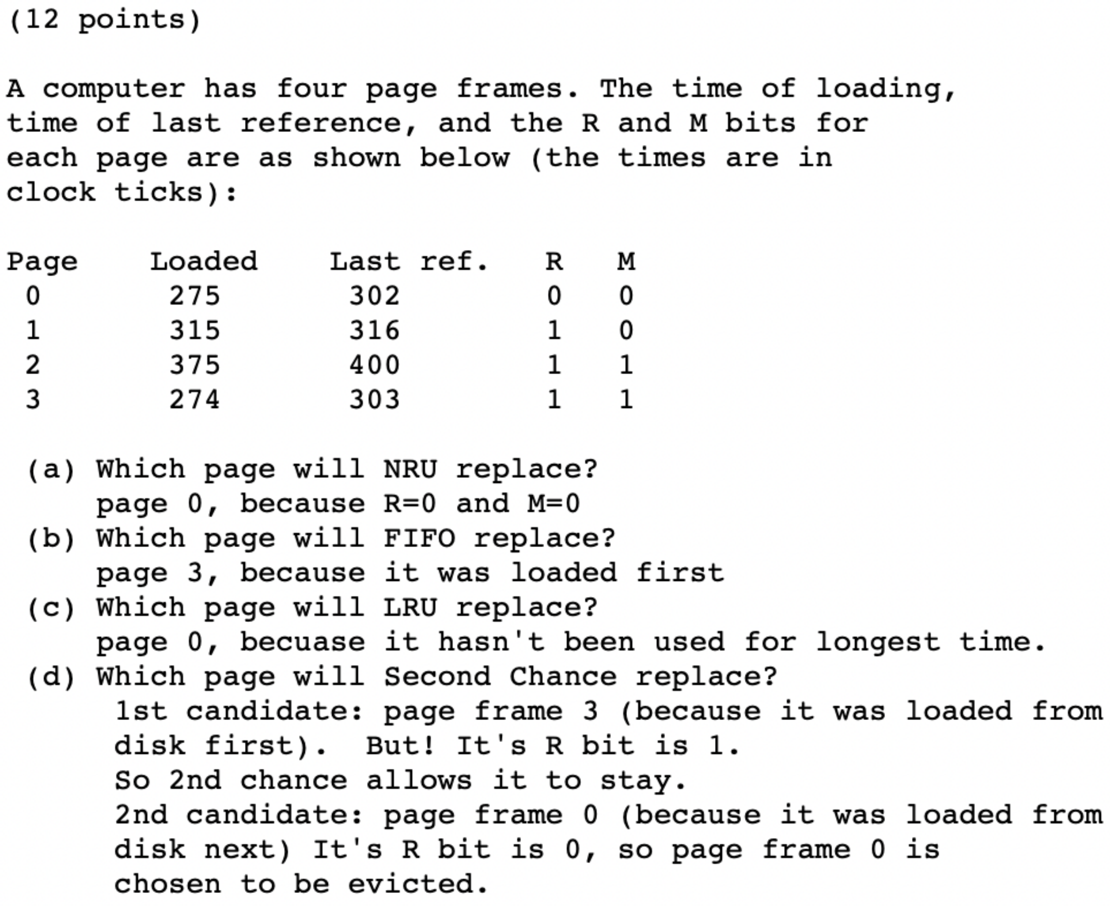
31. A computer has four page frames. The time of loading, time of last access, and the *R* and *M* bits for each page are as shown below (the times are in clock ticks):

Page	Loaded	Last ref.	R	M
0	126	280	1	0
1	230	265	0	1
2	140	270	0	0
3	110	285	1	1

(a) Which page will NRU replace?
(b) Which page will FIFO replace?
(c) Which page will LRU replace?
(d) Which page will second chance replace?

- **NRU** removes Page 2 since it checks R bits and M Bits. If R and M bits both zero, it choses the remove
- **FIFO** removes page 3 since it means first in first out, We check loaded information. page 3 loaded first, so removed first.
- **LRU** removes page 1, since LRU means least recently used, We check last ref attribute and page 1 has the latest accessed page.
- **Second Change** removes page 2, because it works like FIFO but checks the R bit. It controls first added page, if R value is 1, checks the next added page till to found a page with R value 0.

İnternette bu sorunun şöyle bir benzerini de buldum :



(12 points)

A computer has four page frames. The time of loading, time of last reference, and the R and M bits for each page are as shown below (the times are in clock ticks):

Page	Loaded	Last ref.	R	M
0	275	302	0	0
1	315	316	1	0
2	375	400	1	1
3	274	303	1	1

(a) Which page will NRU replace?
page 0, because R=0 and M=0
(b) Which page will FIFO replace?
page 3, because it was loaded first
(c) Which page will LRU replace?
page 0, becuase it hasn't been used for longest time.
(d) Which page will Second Chance replace?
1st candidate: page frame 3 (because it was loaded from disk first). But! It's R bit is 1.
So 2nd chance allows it to stay.
2nd candidate: page frame 0 (because it was loaded from disk next) It's R bit is 0, so page frame 0 is chosen to be evicted.

File System

a) How can we find missed block in file system?

- Answer

b) A file system uses X KB disk blocks. There are a total of 2^10 files on this system. We use bitmap based free block representation. The disk size is Y KB. What would be the wasted disk space for fragmentation and free block representation?

- Answer

c) What is SSD's readability problem?

özel, özgün

- For instance, SSDs have peculiar properties when it comes to reading, writing, and deleting. In particular, each block can be written only a limited number of times, so great care is taken to spread the wear on the disk evenly.

Input/Output

a) What is cylinder skew? What is double interleaving? Show with figures.

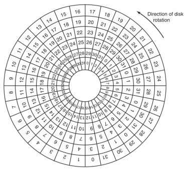


Figure 5-22. An illustration of cylinder skew.

- Cylinder skew: It takes time for reading head of the disk to move from a track to another one. So, if the sectors with same numbers are on top of each other, the reading head would miss it and it would have to wait for a whole spin. In order the avoid this, each track starts with a offset.

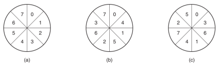


Figure 5-23. (a) No interleaving. (b) Single interleaving. (c) Double interleaving.

If the copying process is very slow, the **double interleaving** of Fig. 5-24(c) may be needed. If the controller has a buffer of only one sector, it does not matter whether the copying from the buffer to main memory is done by the controller, the main CPU, or a DMA chip; it still takes some time. To avoid the need for interleaving, the controller should be able to buffer an entire track. Most modern controllers can buffer many entire tracks.

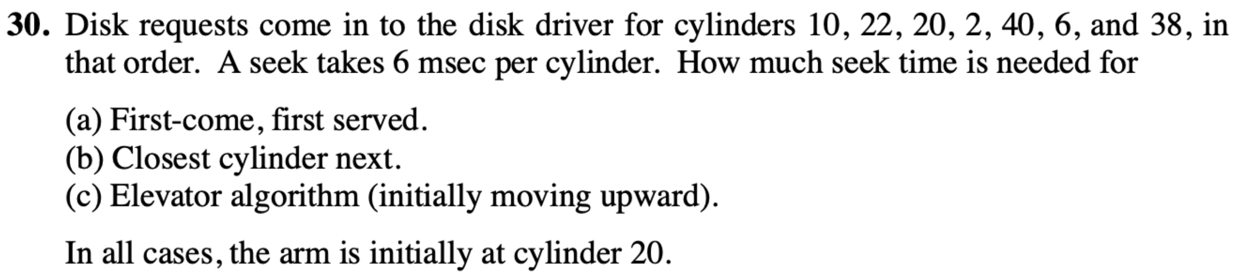
(Tam emin değilim) Double interleaving: After reading a sector, it takes some time to get ready for next read. So, sectors are placed with 2 offset to avoid this problem (0-2-4-1-3 gibi)

b) If a disk has double interleaving, does it also need cyclinder skew in order to avoid missing data when making a track to track seek? Why?

Maybe yes and maybe no. Double interleaving is effectively a cylinder skew of two sectors. If the head can make a track-to-track seek in fewer than two sec- tor times, than no additional cylinder skew is needed. If it cannot, then additional cylinder skew is needed to avoid missing a sector after a seek.

track=tamamı,
sector= onun parçası

c) Bu tarz bir soru çıkmış anladığım kadarıyla :



30. Disk requests come in to the disk driver for cylinders 10, 22, 20, 2, 40, 6, and 38, in that order. A seek takes 6 msec per cylinder. How much seek time is needed for

(a) First-come, first served.
(b) Closest cylinder next.
(c) Elevator algorithm (initially moving upward).

In all cases, the arm is initially at cylinder 20.

- a) 10-22-20-2-40-6-38 (total seek time: 10*seektime + 12*seektime + 2*seektime + 18*seektime + 38*seektime + 34*seektime + 32*seektime = 146*seektime)
- b) 20-22-10-6-2-38-40 (total seek time: 0*seektime + 2*seektime + 12*seektime + 4*seektime + 4*seektime + 36*seektime + 2*seektime = 60*seektime)
- c) 20-22-38-40-10-6-2 (total seek time: 0*seektime + 2*seektime + 16*seektime + 2*seektime + 30*seektime + 4*seektime + 2*seektime = 56*seektime)

2018 - Büt

General

a) Is it important for a user to know if a library function call would make a system call? Explain why or why not.

- (Tam emin değilim) It is not, since it is OS’s job to trap system calls, the end user can be abstracted from this process.

A TLB is a hardware cache that stores recently accessed virtual-to-physical address translations. It helps in accelerating the address translation process by avoiding the need to access the page table in memory for every memory access. When a context switch occurs, the TLB needs to be flushed, as the address space of the process being switched out is no longer valid. This flushing operation takes time and adds to the context switch overhead.

b) Is context switching more expensive or less expensive for systems with no TLB? How about systems with no memory cache? Explain your answer.

- Data transferi daha az oluyo cache olunca, bundan dolayı da daha az daha rahat context swith oluyo

Systems without a TLB or a memory cache experience higher context switch overhead because they lack the hardware acceleration and caching mechanisms that help expedite the address translation and memory access operations. These systems rely more heavily on memory accesses and incur additional delays when switching between processes, leading to increased context switching costs.

c) What are the differences between a device driver and device controller?

- A device driver allows a computer to interface and interact with a specific hardware device, such as a printer, sound card, graphics card, etc. The device controller receives the data from a connected device, stores it temporarily, and then communicates the data to its device driver.

d) What are the main differences between main frame operating systems and personal computer operating systems?

- Mainframes typically run on large boxes with many processors and tons of storage, as well as high-bandwidth busses. PCs are desktop or mobile devices with a single multi-core processor and typically less than 32GB of memory and a few TBs of disk space. Second, a mainframe OS usually supports many simultaneous users.

Processes

a) What are the advantages of monitors? How does Java support monitors?

- a monitor is a synchronization technique that enables threads to mutual exclusion and the wait() for a given condition to become true.
- In Java terminology a monitor is **a mutex lock which is implicitly associated with an object**. When the synchronized keyword is applied to classes or methods an implicit mutex lock is created around the code, which ensures that only one thread at a time can execute it. This is called a monitor lock or just a monitor

b) Assume we are using round robin scheduler on a single CPU system. The context switch time is 5 msecs. We have 10 interactive processes and 10 batch

processes running on this system. We want our interactive processes to respond in at least 200 msecs for user requests. What would be the quantum time for the interactive processes and batch processes? What would be the system efficiency? Explain your formulations and provide all details.

- Answer

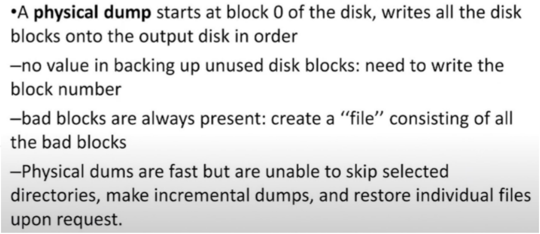
c) **Write a Java Semaphore Class that implements up and down methods. You may use a Mutex class member in your class.**

```
public class Semaphore
{
  private int value;
  // mutex m;
 public Semaphore(int initial){
    this.value = initial;
  }
  synchronized public void up{
    // m.lock();
    value++;
    notify();
    // m.unlock();
  }
  synchronized public void down() throws InterruptedException{
    // m.lock();
    while (value <= 0) wait();
    --value;
    // m.unlock();
  }
}
```

File Systems

a) **What are the advantages and disadvantages of pyhsical and logical dumps in taking disk backups?**

- Advantages of physical dump: It doesn’t do any checks and simply dumps a disk to another one. So, it is much faster (başka avantajı da olabilir hatırladığım bu var)
- Disadvantages of physical dump: It also takes backup of the junk data. Also, it is not possible to recover some part of the disk
- Advantages of the logical dump: It doesn’t take backup of the junk data and temporary files. Also, you can recover only the files you want.

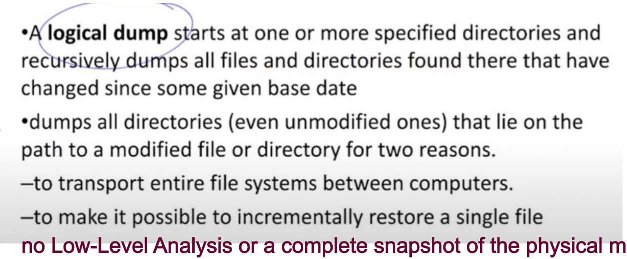


•A **physical dump** starts at block 0 of the disk, writes all the disk blocks onto the output disk in order

–no value in backing up unused disk blocks: need to write the block number

–bad blocks are always present: create a ‘‘file’’ consisting of all the bad blocks

–Physical dums are fast but are unable to skip selected directories, make incremental dumps, and restore individual files upon request.



•A **logical dump** starts at one or more specified directories and recursively dumps all files and directories found there that have changed since some given base date

•dumps all directories (even unmodified ones) that lie on the path to a modified file or directory for two reasons.

–to transport entire file systems between computers.

–to make it possible to incrementally restore a single file

no Low-Level Analysis or a complete snapshot of the physical memory or storage

- Disadvantages of the logical dump: Slow[redacted]

b) A file system uses X-KB disk blocks. There are total of 2^10 files on this system. We use bitmap based free block represantations. The disk size is Y-KB. What would be the wasted disk space for fragmentation and free block represantation? Explain your answer.

- Answer

c) List at least three methods to increase the file system performance and explain them.

- 1- Caching: The most common technique used to reduce disk accesses is the block cache
  or buffer cache.
- 2- Block Read Ahead: A second technique for improving perceived file-system performance is to try
  to get blocks into the cache before they are needed to increase the hit rate.
- 3- Reducing Disk-Arm Motion: Caching and read ahead are not the only ways to increase file-system performance. Another important technique is to reduce the amount of disk-arm motion
  by putting blocks that are likely to be accessed in sequence close to each other

Memory

a) What type of information is stored in a page table entry? Explain each.

- **Frame Number :** It gives the frame number in which the current page you are looking for is present.
- **Present/Absent Bit :** Present or absent bit says whether a particular page you are looking for is present or absent. In case if it is not present, that is called Page Fault.
- **Protection Bit :** Protection bit says that what kind of protection you want on that page.(read, write, execute)
- **Referenced Bit :** Referenced bit will say whether this page has been referred in the last clock cycle or not.
- **Caching Enabled/Disabled Bit :** Some times we need the fresh data. Let us say the user is typing some information from the keyboard and your program should run

according to the input given by the user. In that case, the information will come into the main memory. Therefore main memory contains the latest information which is typed by the user. Now if you try to put that page in the cache, that cache will show the old information. So whenever freshness is required, we don't want to go for caching or many levels of the memory.The information present in the closest level to the CPU and the information present in the closest level to the user might be different. So we want the information has to be consistency, which means whatever information user has given, CPU should be able to see it as first as possible. That is the reason we want to disable caching. So, this bit **enables or disable** caching of the page.

- **Modified Bit :** Modified bit says whether the page has been modified or not. Modified means sometimes you might try to write something on to the page. If a page is modified, then whenever you should replace that page with some other page, then the modified information should be kept on the hard disk or it has to be written back or it has to be saved back.

b) **A computer has four page frames which are empty. If the page references are 0R(page 0 read), 3W(page 3 write), 7W, 2R, 7R, 0W, 2R, 3R, 5W, 6R, 2W , explain which pages are replaced by NRU, FIFO, LRU and second chance algorithms. Give your details.**

- Answer Üstte buna benzer daha düzgün soru var.

c) **A computer with a ~~28~~ 32-bit address uses a two-level page table. Virtual addresses are split into a ~~8~~ 9-bit top-level page table field, an ~~7~~ 11-bit second-level page table field, and an offset. The physical memory can hold 2^24 bytes. We have a process (A) that needs 2^20 bytes and another process (B) that needs 2^10 bytes. Draw a figure to show the memory setup. How large should be the first and the second level pages for these processes?**

- Offset = 32 – (9 + 11) = 12
  Page Size = 2^12 = 4 . 2^10 = 4 KB and 2'20 pages

Input/Output

a) **Give one case where programmed I/O is faster than DMA.**

- When transferred data is only few bytes, than programmed I/O can be faster.

- But if we have lots of data, then DMA is much much faster than programmed I/O

- Because Programmed IO would require the CPU to check continuously for available data or IO port idle, which is a colossal waste of resources.

b) **If a disk has double interleaving, does it also need cylinder skew in order to avoid missing data when making a track-to-track seek? Why?**

- (Tam emin değilim) Yes. Since interleaving is done on the same track (between sectors) and cylinder skew in done between tracks.
- (baska bi cevap) Yes. Since cylinder skew is for avoid missing data when follow sequential data.

c) **What are the advantages and disadvantages of soft timers?**

- Advantages
    - Don't need extra interrupts
- Disadvantages
    - not exact time, only approximate
    - may be not good for control tasks

Soft timers can be used to avoid interrupts and reduce context switches associated with network processing without sacrificing low communication delays.

2019 - Final

a) **What is Timer Device ? Where OS uses it?**

- A hardware ***timer*** is ***used*** to measure and count external events and to carry out other tasks while the ***timer*** process runs.
- (Emin degilke) When a processes runs to long, OS detects it with timer. Os could took some actions to this process.

Sleep and Power Management, task scheduling and multitasking

b) **What are the similarities and differences between Traps and Interrupts?**

- **Trap,** is a signal raised from an instruction(i.e. printf) inside a user program. It indicates the OS to perform on some functionality.
- Traps are synchronous.
- For example calling printf (“%s\n”, str) will invoke the write function to print the output to the standard output. This will invoke a trap and it will pass the control to the trap handler. Then, the user mode changes to kernel mode and the OS

executes the write call. After completing the task, the control is transferred back to the user mode from the kernel mode.

- **Interrupt,** is a signal to the processor created by hardware devices.
- interrupts are asynchronous.

c) In which scheduler, starvation occurs?

- **Starvation** or **indefinite blocking** is a phenomenon associated with the Priority scheduling algorithms, in which a process ready for the CPU (resources) can wait to run indefinitely because of low priority.
- A solution to this problem is **aging**. **Aging** is a technique of gradually increasing the priority of processes that wait in the system for a long time.

d) While POSIX has 100 syscalls, why Windows API has 1000?

- Because windows has GUI, and it comes with more syscalls.

e) Why multithreading is more efficient than multiprocessing? Explain.

- Threads run on the same address space. So there is no need for cache invalidation, reloading all the data again and again etc unlike multiprocessing.

f) What is the cause of trashing? How does operating system detect trashing?

- Reason: Memory is full, so the pages that are currently in use are removed in order to make room for new ones. So, page faults occur over and over again at each instruction.
- How does OS detect trashing: (sanırım) with mesuring the PFF(page fault frequency). So os can see that there are page replacing over and over.

g) What is RAID? Design a level of RAID that are good for disk read speed and another level that are good for disk crashes. Show your solutions with figures.

- RAID is for using parallelisation for reading disks faster.level 1 disk crashes için en uygunu.

2020 - Makeup

General OS

a) What is the protection hardware? Is it always needed for all OS types?

- Hardware security is vulnerability protection that comes in the form of a physical device rather than software that is installed on the hardware of a computer system. Hardware security can pertain to a device used to scan a system or monitor network traffic. Common examples include hardware firewalls and proxy servers.

- (Emin degilim) Her os için gerekli değil, bi diseign Chocie.su

b) What is microkernel? Does a microkernel have more or less system procedure calls compared to POSIX competible system? less purpose is that

- Microkernel is a kernel implementation which keeps only the required OS stuff in the kernel and putting the rest in the user space. This makes the kernel more error proof since one error on kernel can crash the whole system.

c) What are time multiplexing and space multiplexing? Give one example for both?

- Space-multiplexed means that the resource can be divided into two or more distinct units of the resource.(example : an analogue stereo audio cable, with one pair of wires for the left channel and another for the right channel,)

- Time multiplexed means a resource is not divided into units. Instead, a program is allocated exclusive control of the entire resource for a short period of time. (example : The GSM telephone system)

d) What are the main design goal differences between main frame operating systems and mobile computer operating system?

- Mainframes typically run on large boxes with many processors and tons of storage, as well as high-bandwidth busses. PCs are desktop or mobile devices with a single multi-core processor and typically less than 32GB of memory and a few TBs of disk space. Second, a mainframe OS usually supports many simultaneous users.

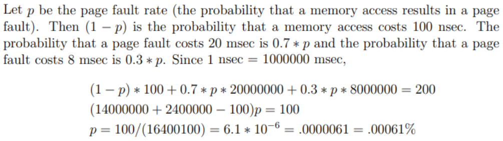
Memory

a) Compare paging with segmentation with respect to the amount of memory required by the address translation structures in order to convert virtual addresses to physical addresses.

- Answer

b) Assume we have a paged virtual memory. The page table is held in registers. It takes 8 milliseconds to service a page fault if an empty page is available or the replaced page is not modified, and 20 milliseconds if the replaced page is modified. Memory access time is 100 nanoseconds. Assume that the page to be replaced is modified 70 percent of the time. What is the maximum acceptable page-fault rate for an effective access time of no more than 200 nanoseconds? Show your work.

- Answer



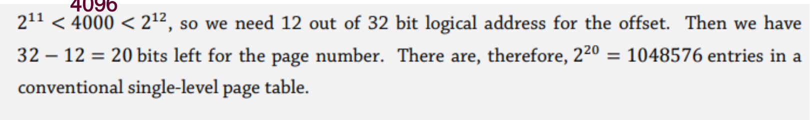
Let p be the page fault rate (the probability that a memory access results in a page fault). Then $(1-p)$ is the probability that a memory access costs 100 nsec. The probability that a page fault costs 20 msec is $0.7 * p$ and the probability that a page fault costs 8 msec is $0.3 * p$. Since 1 nsec = 1000000 msec,

$$(1-p) * 100 + 0.7 * p * 20000000 + 0.3 * p * 8000000 = 200$$
$$(14000000 + 2400000 - 100)p = 100$$
$$p = 100/(16400100) = 6.1 * 10^{-6} = .0000061 = .00061\%$$

c) Consider a computer system with a 32-bit logical address and 4-KB page size. The system supports up to 512 MB of physical memory. What is the page table size in terms of bytes if we assume each page table entry is 32 bytes?

- Answer

4096



$2^{11} < 4000 < 2^{12}$, so we need 12 out of 32 bit logical address for the offset. Then we have $32 - 12 = 20$ bits left for the page number. There are, therefore, $2^{20} = 1048576$ entries in a conventional single-level page table.

Input/Output

a) SSD type storage devices don't have any disk arm motion optimization problems. What are the main optimization problems for SSD's in terms of system reliability.

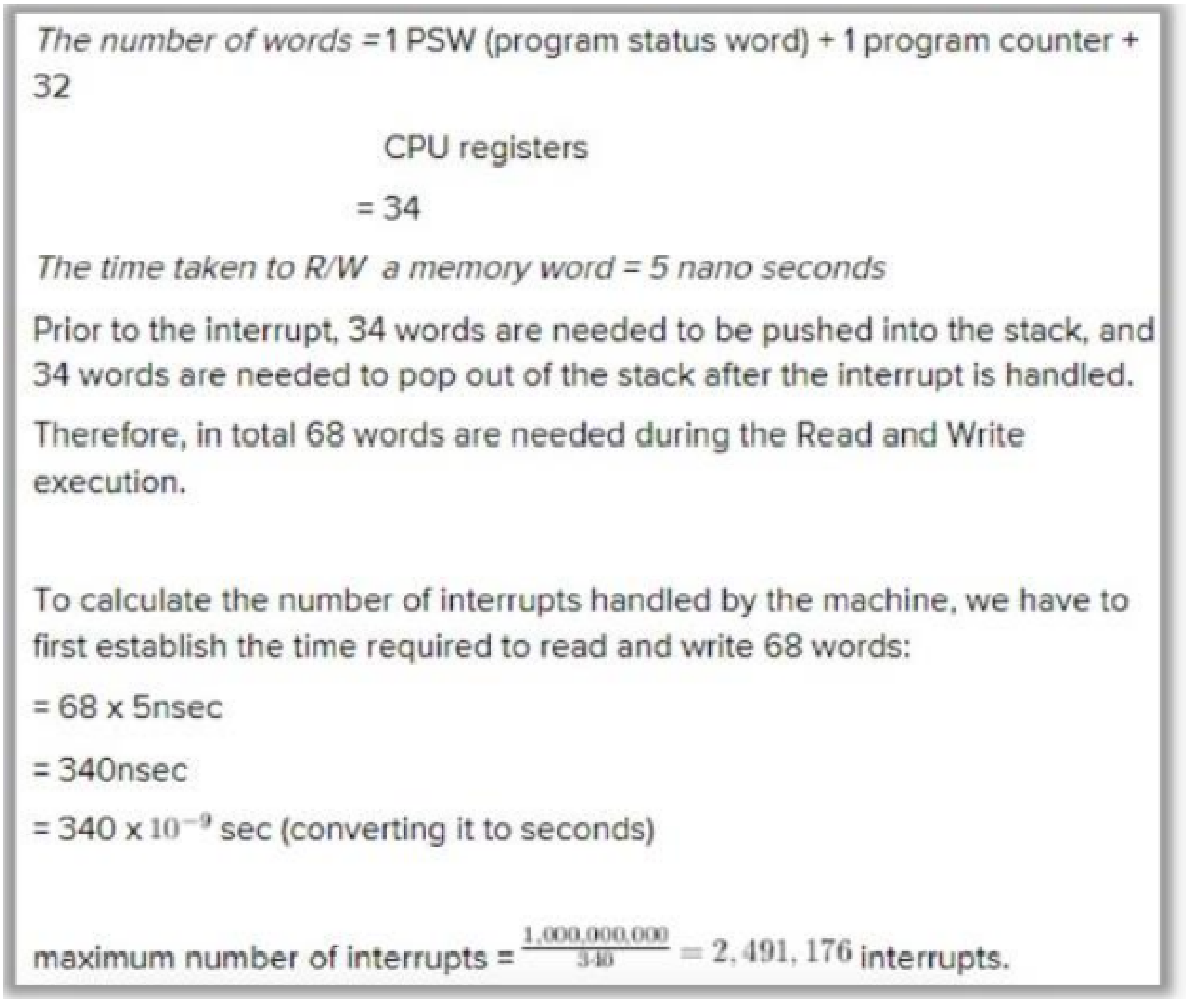
SSDs have a limited lifespan in terms of the number of write/erase cycles they can endure before their performance and reliability degrade.

SSDs require a process called garbage collection to reclaim space from previously used but now invalid data.

- Answer

b) Suppose that a computer can read or write a memory word in 5 sec. Also suppose that when an interrupt occurs, all 32 CPU registers, plus the program counter and PSW are pushed onto the stack. What is the maximum number of interrupts per second this machine can process if we want to be at least 50% efficient?

An interrupt requires pushing 34 words onto the stack. Returning from the interrupt requires fetching 34 words from the stack. This overhead alone is 340 nsec. Thus the maximum number of interrupts per second is no more than about 2.94 million, assuming no work for each interrupt.



The number of words = 1 PSW (program status word) + 1 program counter + 32

CPU registers

= 34

The time taken to R/W a memory word = 5 nano seconds

Prior to the interrupt, 34 words are needed to be pushed into the stack, and 34 words are needed to pop out of the stack after the interrupt is handled.

Therefore, in total 68 words are needed during the Read and Write execution.

To calculate the number of interrupts handled by the machine, we have to first establish the time required to read and write 68 words:

= 68 x 5nsec

= 340nsec

= 340 x 10^{-9} sec (converting it to seconds)

maximum number of interrupts = $\frac{1,000,000,000}{340} = 2,491,176$ interrupts.

c) What is RAID? Design a level of RAID that are good for disk read speed and another level that are good for disk crashes. Show your solutions with figures.

- Answer

d) Consider a system running ten I/O-bound tasks and one CPU-bound task. Assume that the I/O-bound tasks issue an I/O operation once for every millisecond of CPU computing and that each I/O operation takes 10 milliseconds to complete. Also assume that the context switching overhead is 0.1 millisecond and that all processes are long-running tasks. Calculate the quantum time to achieve a CPU utilization greater than 94%. Show all your work.

- Answer

2021 - Final

Question 1

a) Explain why some users liked first generation computers more than second generation computers. What was the solution for these users?

First Generation Computers

- First generation computers used vacuum tubes.
- All programming done in machine language or even worse yet, by wiring up electrical circuits by connecting thousands of cables to plugboards to control the machine’s basic functions.
- There were no OS, programming language, anything but wires and operators(people).
- It weighted tens of tons and was very expensive

Second Generation Computers

- Instead of vacuum tubes, transistors began to be used.**(Mainframe)**
- Languages like Assembly and FORTRAN came out so programming became more easy.
- Disadvantage of second generation computers was, they were producing a lot of heat
- Punch cards used to program.
- First anchestors of operating system seen in this generation

b) Which one has more system calls: UNIX or Windows? Why?

- Win > Unix, due to Win API’s function call.

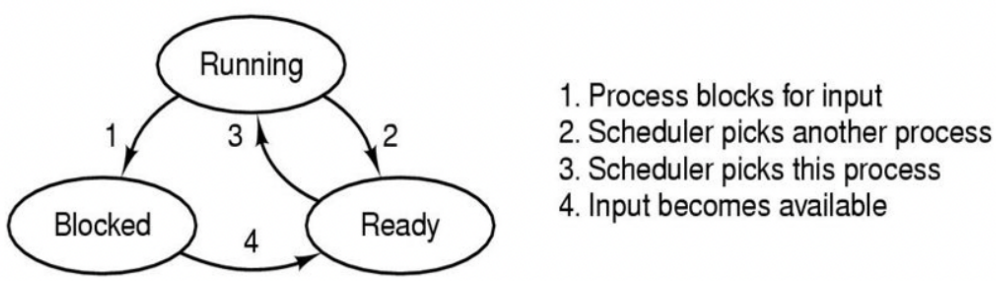
c) What are the advantages and disadvantages of keeping the kernel size small?

- **Kernel** is central component of an operating system that manages operations of computer and hardware.
- It basically manages operations of memory and CPU time. It is core component of an operating system.

- **Advantage of small size kernel :** It is more stable than others (Monolithic vs Microkernel)
- **Disadvantage of small size kernel :** Leads to more system calls and context switches.

Question 2

a) Explain the 4 thread transition cases and give example for each. (Sınavda aşağıdaki gibi görsel verilmiş)


Running
Blocked
Ready
1
2
3
4
1. Process blocks for input
2. Scheduler picks another process
3. Scheduler picks this process
4. Input becomes available

- **Running → Blocked :** Running thread goes to blocked state when it asks for asynchronous task(For example getting data from the disk).
- **Running → Ready :** When a program calls thread_yield() function or when an interrupt happens(or any context switch decided by os) running thread goes to ready state. In this state, threads wait in a queue to go back to running state.
- **Ready → Running :** If thread is at start of the queue and running thread is interrupted for some reason, then thread waiting at ready state goes to running state and starts using CPU.
- **Blocked → Ready :** After asynchronous task is completed(data arrived from disk), blocked thread goes to ready state.

c) Explain why the hardware saves only a small number of registers instead of saving all of them when a interrupt happens.

- Because saving saving all registers take more time, occupies more memory(also unnecessary). All we have to do is, saving the registers that running process use and modify.

Question 3

a) How would make the fork operation more efficient if we know that the child process after the fork makes a execute system call most of the time?

- Answer

b) An I-node system has root dir, it includes 3 dir (dir1, dir2, dir3). dir1 include 2 file (file1, file2). Draw possible inode, block, dir configuration for this file system. Chose your inode numbers and block numbers yourself.

- Answer

c) Explain why we do not prefer LRU for page replacement while it is very popular for disk caches. Describe an efficient way of using LRU for disk cache management.

- (emin değilke) There are two types of LRU. Hardware assistant type requires an hardware, and software asistant types is slower than the hardware one. WSClock is the best option for page replacement.
- İlk cevaba ek olarak, LRU ne olursa daha iyi olurun cevabını verirsek tam olur bu s

Question 4

a) Why does low level formatting take more time than other types? Give examples both for UNIX and Windows.

- Low-level formatting refers to making no assumptions about what is already there and creates all partitions and etc from scratch. All sectors might be checked and initialized in the process which can take a long time.
- High-level formatting is a faster mechanism and basically just empties an already established file system and may rely on assumptions about what is already there.

b) How do you do the virtualization of the CPU if do not have CPU based virtualization support?

- A CPU architecture is virtualizable if it supports the ability to run the VM's privileged and unprivileged instructions in the CPU's user mode.

c) List the things that your hardware and OS do when a timer interrupt happens. Do not forget about your process/threads, virtual memory etc.

- First, CPU saves the current state of the process/threads and it jumps to the interrupt handler code. Meaning interrupt handler stores the program counter and the status register in memory, along with the contents of any data registers it plans to use.
- Second, the interrupt controller passes the interrupt number to the CPU
- Then, CPU uses this number to search an interrupt vector table to find the appropriate interrupt service routine to execute
- Finally it calls interrupt service routine and after finishing it, resumes the process.