1:53

Name: _ ATS account: icsc.edu 69

Part I: Multiple Choice

Each question is worth 2 points (there are 40 questions). Indicate your choice by putting an **X** in the appropriate circle.

1. *(2 points)* In the process state transition diagram, the transition from the READY state to the RUNNING state indicates that:

50

- A. A process was preempted by another process.
- B. A process has blocked for a semaphore or other operation.
- C. A process is done waiting for an I/O operation.
- D. A process was just created.

2. *(2 points)* Which of the following is shared between all of the threads in a process? Assume a kernel-level thread implementation.

- A. Register values
- B. File descriptors
- C. Scheduler priority
- D. Local variables

3. *(2 points)* Which of the following is *not* true?

- A. Shortest Remaining Time Next is the best preemptive scheduling algorithm in terms of turnaround time.
- B. Priority scheduling can suffer from starvation.
- C. Lottery scheduling is preemptive.
- D. Multi-level feedback queues guarantee equal time to all processes.

4. *(2 points)* A critical region is:

- A. The part of a program in which shared data is accessed.
- B. The most important part of a program.
- C. The part of the kernel that interfaces directly to the device controllers.
- D. The part of a program in which a bug would cause the program to exit.

5. *(2 points)* Which of the following is *not* used for synchronization?

- A. The banker's algorithm
- B. The bakery algorithm
- C. Busy waiting with test and set
- D. Monitors

6. *(2 points)* Which of the following is *not* true of virtual memory?

- A. It allows more efficient use of memory.
- B. It requires hardware support.
- C. It reduces the need for relocatable code.
- D. It requires the use of a disk or other secondary storage.

7. *(2 points)* Which of the following is not usually stored in a two-level page table?

-4

- A. Virtual page number
- B. Physical page number
- C. Dirty bit
- D. Reference bit

Name: ATS account .ucsc.edu

8. *(2 points)* Which of the following paging algorithms is most likely to be used in a virtual memory system?
- ○ A. FIFO
- ○ B. Second chance
- ○ C. Least Recently Used
- ○ D. Least Frequently Used

9. *(2 points)* The purpose of a TLB is:
- ○ A. To cache page translation information
- ○ B. To cache frequently used data
- ○ C. To hold register values while a process is waiting to be run
- ○ D. To hold the start and length of the page table

10. *(2 points)* Which of the following is *not* true about segmented memory management?
- ○ A. Segment length must be a multiple of the page size. ✓
- ○ B. Segmentation allows multiple linear address spaces in one process.
- ○ C. Segmentation can be used with paging to keep segments partially resident in memory.
- ○ D. A segment can be read-only for one process and read-write for another.

11. *(2 points)* System calls:
- ○ A. Provide a rich and flexible API for software developers.
- ○ B. Often change dramatically between different releases of an operating system.
- ○ C. Protect kernel data structures from user code.
- ○ D. Allow the operating system to optimize performance.

12. *(2 points)* What is the main difference between traps and interrupts?
- ○ A. How they are initiated
- ○ B. The kind of code that's used to handle them
- ○ C. Whether or not the scheduler is called
- ○ D. How the operating system returns from them

13. *(2 points)* Buffering is useful because
- ○ A. It makes it seem like there's more memory in the computer.
- ○ B. It reduces the number of memory copies required.
- ○ C. It allows all device drivers to use the same code.
- ○ D. It allows devices and the CPU to operate asynchronously.

14. *(2 points)* The main advantage of DMA is that it
- ○ A. Increases system performance by increasing concurrency.
- ○ B. Allows the CPU to run faster.
- ○ C. Reduces the traffic on the data bus.
- ○ D. Removes the requirement that transfers be properly aligned.

-2

15. *(2 points)* Which of the following disk seek algorithms would be the best choice to implement in a system that services an average of 5 disk requests per second?
- ○ A. First Come First Served
- ○ B. Shortest Seek Time First
- ○ C. SCAN X
- ○ D. C-SCAN X

Name: 'S account: ucsc.edu

16. *(2 points)* Which of the following disk seek algorithms has the *most* variability in response time?

- A. First Come First Served
- B. Shortest Seek Time First
- C. SCAN
- D. C-SCAN

17. *(2 points)* A typical hard drive has a peak throughput of about

- A. 2×10^5 bytes per second.
- B. 2×10^6 bytes per second.
- C. 2×10^7 bytes per second.
- D. 2×10^8 bytes per second.

18. *(2 points)* RAID is a way to:

- A. Increase hard drive latency and performance.
- B. Increase hard drive performance and decrease cost.
- C. Increase hard drive reliability and performance.
- D. Increase hard drive reliability and decrease cost.

19. *(2 points)* Which of these would *not* be a good way for the OS to improve battery lifetime in a laptop?

- A. Shut down the hard drive until it's needed.
- B. Reduce the processor speed while it's idle.
- C. Turn off power to the memory.
- D. Shut down the modem when it's not connected.

20. *(2 points)* Which of the following is *not* included in an inode in Linux?

- A. File owner
- B. File name
- C. File modification date
- D. Pointer to the first data block

21. *(2 points)* Which of the following is *not* true of a file system using a file allocation table (FAT)?

- A. A bitmap is used to track free blocks.
- B. Index nodes (inodes) are unnecessary.
- C. File size is limited only by the amount of free space.
- D. Block n of a file can be read without first reading blocks 0 through $n-1$.

22. *(2 points)* Which of the following file system types requires the *fewest* 4 KB file blocks to store a 16 KB file? Don't count space used by the inode, if any.

- A. One-level (single indirection) indexed allocation
- B. Linked allocation
- C. Contiguous allocation
- D. They all require the same number of blocks

23. *(2 points)* Given a block size of 4 KB and a disk that holds 100 GB of data (using 32 bit disk addresses), how big can a file be if you have a single index block in an indexed file management scheme?

- A. 4 KB
- B. 4 MB
- C. 16 MB
- D. 100 GB

Name: CATS account: csc.edu

24. *(2 points)* What purpose does a journal serve for a file system?

- A. Contains all of the metadata structures for the file system.
- B. Records all reads and writes performed on the file system.
- C. Ensures that on-disk structures remain consistent.
- D. Guarantees that data written to disk is never lost.

25. *(2 points)* Which of the following is true of memory-mapped files?

- A. Memory-mapped files can be used for process synchronization.
- B. It's impossible to increase the size of a memory-mapped file.
- C. They eliminate the need to transfer file data to and from disk.
- D. Programs need not make read and write system calls for them.

26. *(2 points)* Which of the following is *not* true about the file system directory hierarchy in Linux?

- A. Directories are stored in the same way as files.
- B. Directory entries are a fixed size.
- C. It can be represented as a directed graph.
- D. A single file can have multiple names in the same directory.

27. *(2 points)* Which of the following is *not* a benefit of using larger file blocks in a file system using indexed allocation (such as Linux)?

- A. Better disk space utilization.
- B. Faster access to a randomly-selected block of a large file.
- C. Higher average transfer rate for file data.
- D. Less disk space consumed for metadata structures.

28. *(2 points)* Which of the following is an advantage for having disk quotas?

- A. It ensures that the free space on a disk is distributed equitably to all users.
- B. It stops a malicious process from filling the disk.
- C. It prevents a process from consuming too much disk bandwidth.
- D. It makes it easier to group a user's files together.

29. *(2 points)* In UNIX (or DLXOS), how many system calls must a process make to create a new file, write three blocks to it, and close it?

- A. One
- B. Three
- C. Four
- D. Five

30. *(2 points)* The three main goals of computer security are:

- A. Detection, response, and correction
- B. Confidentiality, performance, and reliability
- C. Confidentiality, integrity, and availability
- D. Detection, protection, and access control

31. *(2 points)* Which of these statements about public-key encryption is false?

- A. Public-key encryption doesn't require a secret key
- B. Public-key encryption can be used to authenticate the sender of a message.
- C. Public-key encryption is slower than symmetric-key (shared-key) encryption.
- D. Decryption and encryption are the same function for public-key encryption.

Name: ATS account: _ csc.edu

32. *(2 points)* Moving from a 64-bit key to a 128-bit key for symmetric-key encryption makes brute-force decryption (guessing the key) how much more difficult?

- ○ A. $2\times$ harder
- ○ B. $64\times$ harder
- ○ C. $64^2\times$ harder
- ○ D. $2^{64}\times$ harder

33. *(2 points)* One-way functions are used in:

- ○ A. Public-key encryption
- ○ B. Symmetric-key encryption
- ○ C. Authentication
- ○ D. Permission revocation

34. *(2 points)* The Linux file system uses a form of which protection mechanism?

- ○ A. Access control lists
- ○ B. Capabilities
- ○ C. Protection domains
- ○ D. Security by obscurity

35. *(2 points)* A trojan horse is:

- ○ A. A program that does something beyond its apparent purpose.
- ○ B. A program that shuts down if certain conditions are not met.
- ○ C. A special password that's built into the system software.
- ○ D. A computer virus that's spread by email.

36. *(2 points)* Which of the following is *not* a good technique for ensuring that passwords are not compromised?

- ○ A. Storing a hashed version of passwords.
- ○ B. Forcing users to pick passwords that are not in the dictionary or other lists of common words.
- ○ C. Adding salt to the password file.
- ○ D. Storing the password file where only the operating system can read it.

37. *(2 points)* Which of the following is *not* a technique often used to break into computer systems?

- ○ A. Exploiting buffer overflows
- ○ B. Replaying login sessions
- ○ C. Hijacking the compiler
- ○ D. Sending malicious code via email

38. *(2 points)* The DLX code uses which scheduling algorithm by default?

- ○ A. Round robin
- ○ B. First come, first served
- ○ C. Lottery
- ○ D. Priority

39. *(2 points)* How does the vanilla version of DLXOS (before you made changes) keep track of open files?

- ○ A. Global open file table, at most one entry per UNIX file.
- ○ B. Global open file table, multiple entries possible per UNIX file.
- ○ C. Per-process open file table.
- ○ D. Dynamically-allocated structures to keep track of open files.

40. *(2 points)* The DLXOS system call `Spawn()` returns:

- ○ A. The exit status of the child process.
- ○ B. The current status of the child process.
- ⊗ C. The process ID of the child process.
- ○ D. A pointer to the PCB of the child process.

Name: | 'S account: csc.edu

Part II: Short Answer

Each question is worth 12 points (there are 10 questions). Show all of your work—you can get partial credit for partially correct answers, and may not get full credit for answers without any work.

9

1. *(12 points)* **Disk Seek Algorithms** 884, 1764, 4478, 4502, 4823, 5041, 6881, 8793, 9337

A disk request queue has requests for blocks on the following cylinders (ordered by time of arrival):
1764, 5041, 4502, 6881, 884, 4823, 8793, 4478, 9337

The disk has 10,000 cylinders numbered 0 through 9999. The disk head is currently at cylinder 4630 and is moving towards cylinder 9999.

a. *(8 pts)* Calculate the total seek distance for each of the following disk seek algorithms: FCFS, SSTF, LOOK, C-LOOK.

7

FCFS: (1764 + 4630) + (1764 + 5041) + (5041 - 4502) + (6881 - 4502)
+ (6881 - 884) + (4823 - 884) + (8793 - 4823) + (8793 - 4478)
+ (9337 - 4478) = 2876 + 3277 + 539 + 2379
+ 5997 + 3939 + 3970 + 4315 + 4859
= 32,151 cylinders

SSTF: (4630 - 4502) + (4502 - 4478) + (4823 - 4478) + (5041 - 4823)
+ (6881 - 5041) + (8793 - 6881) + (9337 - 8793) + (9337 - 1764)
+ (1764 - 884)
= 128 + 24 + 345 + 218 + 1840 + 1912 + 544 + 7573 + 880
= ~~12,584 cylinders~~ 13,464 ✓ cylinders

LOOK: (4823 - 4630) + (5041 - 4823) + (6881 - 5041) (8793 - 6881) + (9337 - 8793)
+ (9337 - 4502) + (4502 - 4478) + (4478 - 1764) + (1764 - 884)
= ~~12,280 cylinders~~ 13,160 cylinders

C-LOOK: 193 + 218 + 1840 + 1912 + 544 + (9337 - 884) + (1764 - 884)
+ (4478 - 1764) + (4502 - 4478)
C-LOOK: 16,778 cylinders.

b. *(4 pts)* Modern disk drives don't provide an easy way to calculate a cylinder number from a disk block address. How could you adapt the above algorithms to operate without knowing the cylinder on which a given disk block is located?

2

b) To adapt the above algorithms w/out knowing the cylinder on which a given disk is located we can implement a virtual disk where the physical cylinder on which a block is located is mapped to a virtual cylinder on the OS's representation of the disk.

Name: ATS account: _ c.edu

2. *(12 points)* **Deadlock**

a. *(4 pts)* List the four conditions that must hold for deadlock to occur.

10

4

1) No preemption
2) Hold and wait (keeps resource while waiting for more)
3) Circular wait
4) Mutual Exclusion

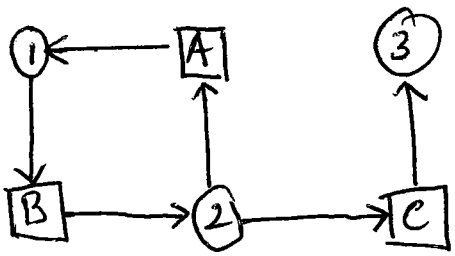
b. *(4 pts)* List the four strategies for dealing with deadlock.

2

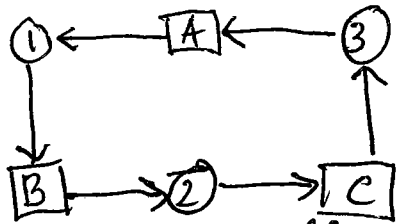
1) Preemption (allowing one process to give up its resource so another can finish)
2) Don't allow a process to hold resource while it waiting for another. (Two phase locking).
3) Identifying ~~poten~~ potential circular waits & prevent them
4) One ~~process~~ resource per process at any moment.

c. *(2 pts)* Draw a resource allocation graph with three processes (1–3) and three resources (A–C) where the two processes are deadlocked.

deadlock on processes 1 & 2. 2



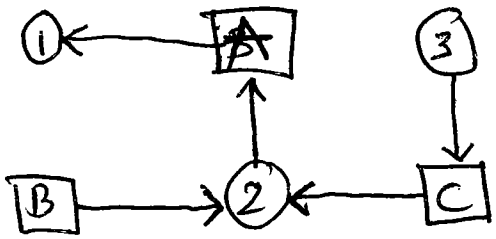
deadlock on all processes



all

d. *(2 pts)* Draw a resource allocation graph with three processes (1–3) and three resources (A–C) where ~~both~~ resources are allocated but the processes are not in deadlock.

2



END OF THE FIRST HALF OF THE EXAM

77

Name: TS account: ucsc.edu

3. *(12 points)* **Memory Management**

Suppose that your memory management hardware supports neither reference bits nor modified (dirty) bits, but does support valid/invalid bits and the ability to make pages read-only or read-write. Explain you could use this hardware to implement a software solution that will emulate the functionality of the missing dirty and reference bits. Your solution should be as efficient as possible.

7

To emulate the functionality of the missing dirty & reference bit using software we can have the software check the valid/invalid bit. When a valid bit is on, we mark the page as not dirty. When ~~a page~~ the valid bit is on and the page allows "read", we mark the page as non-dirty and referenced. When the valid bit is off (invalid) we mark the page as dirty. ~~and give it read-write access.~~ When the invalid bit is off and read-only, we mark the page as dirty and referenced. (1)

Doing this scheme, we have a way of know ~~what data~~ which page was changed and if they haven been referenced ~~off~~ or not.

(1) when the page has read-write access we mark it as dirty and referenced.

Name: ATS account: ucsc.edu

4. *(12 points)* **File Systems**

Describe how to implement a file system consistency checker for Linux that verifies that only free blocks are marked free, only allocated blocks are marked allocated, and each allocated block is allocated to exactly one file.

(10)

Assuming that the file system has a way to keep track of free blocks and allocate blocks. We can have the checker look at the record of free blocks and allocated blocks to find overlapping. If there are overlaps then the fs is not consistent. After that the checker can look at allocated blocks and see which file has a reference to them by following the inodes. If multiple reference were made to the block it is allocated to multiple files.

Another way to make sure one block to one file allocation is to check the allocated block table every time allocating a new block for a file. If a block is not free then don't use it.

Name: TS account: csc.edu

5. *(12 points)* **Security**

Based on your knowledge of UNIX and the things we have discussed in this class, describe two realistic ways that someone might be able to get another person's login information (username and password) for one of the UNIX systems on this campus without just being told the information by anyone (the user or someone else). Don't simply list the names of known attacks—explain how they might actually work. Be as specific as possible.

12

Spoofing Login:
create a program which draws the login screen to ~~the~~ look exactly like that of the sdaris login. This program will be run in the cracker's account. When a user comes to the terminal without knowing this is a fake login screen, he will type in his login name & password which the ~~fake~~ login program will record and save somewhere the cracker can later retrieve. The program then terminates, ~~and the~~ logout the cracker and the real login screen will now run.

login attack:
Create a list of usernames by doing a finger command / or whom command, or look at the list of personal websites on CATS web page.
Create a list of passwords with a dictionary file.
Use this two list and iterate through each usernames and try the passwords in the password file.
Keep trying to ~~to~~ logon to the unix machine through telnet or ssh.

Name: _ S account .ucsc.edu

6. *(12 points)* **File System Snapshots**

(9) Many file systems have the ability to take a *snapshot*—a copy of the entire file system at a particular point in time. After the snapshot is taken, the file system continues to run, with users reading and writing the most recent version of the file system. Something similar is implemented with the `OldFiles` directory on `unix.ic`, which always has a copy of your files as of yesterday.

a. *(4 pts)* List two distinct uses for snapshots.

2 Snapshots can be use with the journal system where the file system can go back to a certain point in time when the file system is not corrupted to recover an filesystem error / crash.

b. *(8 pts)* Briefly describe how you might implement snapshots efficiently. Keep in mind that you want to take snapshots quickly (a few seconds at most) and don't want to keep multiple copies of files that don't change much.

7 Record every write and read operation on the disk. Keep tracks of disk system calls and the parameters passed to these system calls in a log file. ~~like read(), open()~~

Name: _ S account: _ :sc.edu

7. *(12 points)* **Scheduling**

Our operating system is using a scheduling algorithm that gives a higher priority to processes that have used the least CPU time in the recent past.

10

a. *(4 pts)* Why will this algorithm favor I/O-bound processes but not completely starve CPU-bound processes?

this algorithm favors I/O processes because they spend most of their time waiting & thus not using the CPU. So they will be given more priority.

b. *(5 pts)* Give a formula for priority that would result in a good schedule for this algorithm. Your algorithm should allow a process to move from CPU-bound to I/O-bound and back over longer intervals. Assume you have a function *cpu_usage(int i)* that returns the total CPU usage (up to the current time) of process *i*. Hint: to get the CPU time used in the interval $\langle t, t+\Delta \rangle$, you can call this at time t, call it again at $t+\Delta$, and take the difference.

4

$P_i = \text{cpu\_usage}(i_{t_2}) - \text{cpu\_usage}(i_{t_1}) + \frac{1}{2} P_{\text{prev}}$

c. *(3 pts)* What scheduling algorithm would you choose to actually implement this scheduler given the values generated from the previous function? Justify your choice.

3

Using a round robin algorithm which operate on queues of processes for each priority level would work best. This algorithm will run those with highest priorities first.

Name: CATS account .ucsc.edu

(1)

8. *(12 points)* **Segmentation**

The Intel Pentium uses both segmentation and paging to resolve addresses. It uses segmentation to generate a 32-bit virtual address, and uses two-level page tables to translate the virtual address into a physical address.

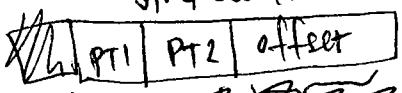
a. *(8 pts)* Show how an address consisting of ⟨*segment_number*, *offset_within_segment*⟩ is translated into a physical address in such a system.

8/6

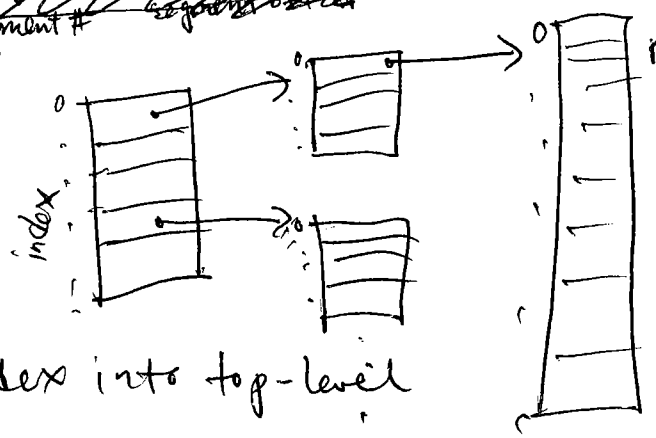
segmentation → virtual address

How?

virtual addr.



~~segment # segment offset~~



PT1 is part

The 32-bit address generated by segmentation is used in a paging scheme to index into the two page table which then produce a physical address. with this address we can add the offset to get to the segment.

PT1 index into top-level table.

PT2 index into second level table.

Part of the 32-bit address is the offset which is used to produce the physical address.

b. *(4 pts)* How does the use of both segmentation and paging reduce memory usage requirements? Are there other benefits to the use of segmentation and paging?

3

Segmentation and paging reduce memory usage because not only blocks of commonly used memory are ~~put into the look up tables~~ mapped into memory but commonly used memory segments are mapped onto memory. Thus reducing the need to look into the memory blocks to find a particular segment.

00,000

Name: CS account: ucsc.edu

9. *(12 points)* **I/O System Performance**

Your computer system has a PCI bus that runs at 500 MB/sec, and two controllers. One controller, which does not support DMA, interrupts the CPU every time a 32-bit word of data is ready. The other controller supports DMA.

a. *(4 pts)* Assume that the CPU takes 20 μs to handle an interrupt, including the transfer of the word of data. How many bytes per second can the OS transfer via the non-DMA controller? How long does it take to transfer a 1 KB disk block using this controller?

11

4 non-DMA: 4 bytes/interrupt $20\mu s = 20 \cdot 10^{-6} s$

$\frac{4 \text{ bytes}}{20 \cdot 10^{-6} s} \times \frac{10^6}{10^6} = \frac{2 \cdot 10^5 \text{ bytes}}{1 \text{ s}} = \frac{2 \cdot 10^5 \text{ bytes}}{\text{second}} = \frac{200{,}000 \text{ bytes}}{\text{second}}$ ≈ 200kB ✓

roughly 200,000 bytes/second.

$\frac{1 \text{ KB}}{200 \text{ KB/s}} = \frac{1}{200}$ second. roughly 1/200 of a second. ✓

b. *(4 pts)* Assume that it takes 200 μs for the operating system to manage the details associated with a DMA request (including any interrupts, but excluding time spent transferring the data itself). If each word transferred by the DMA controller must travel over the PCI bus twice, how large must a request be to make DMA faster than word-by-word transfer?

4 DMA: Since 4 bytes/interrupt. 20μs/interrupts. → 4 bytes/20μs

$\frac{4 \text{ bytes}}{20\mu s} \times \frac{200\mu s}{\text{DMA}} = \frac{40 \text{ bytes}}{\text{DMA}}$.

A transfer must be 40 bytes to equal the transfer rate of non-DMA so a tranfer of any larger than 40 bytes is faster.

c. *(4 pts)* Is there any advantage to allowing the DMA controller to have multiple outstanding requests? Why or why not? How would this affect operating system design?

3 Allowing the DMA controller to have multiple outstandig requests decreases the number of interrupts to the cpu for I/O requests. Spooling is one example of the concept but with software. The OS design must allow DMA drivers.

(9) Name: _____ CATS account: _____ csc.edu

10. *(12 points)*

a. *(6 pts)* Explain how a message is sent from Alice to Bob using public-key encryption (as done in PGP). You should describe any encryption and decryption operations that are performed.

3 Bob has a pair of public & private keys.
The public key is the encryption key. (K_E)
the private key is the decryption key. (K_D).
Alice sends a message to Bob by obtain the public key (K_E). She uses it to encrypt her message, M, using encryption algorithm, E,

$E(K_E, M) \rightarrow C$ Cypher text, C, is produced.

Bob takes the cypher text, C, and uses his private key (K_D) with Decryption algorithm, D, to retrieve the message, M. $D(K_D, C) \rightarrow M$.

b. *(3 pts)* What information (other than the message itself) must Alice and Bob possess for this communication?

3 Alice needs the public key and encryption algorithm.

Bob needs the ~~the~~ private key and decryption algorithm.

c. *(3 pts)* Describe one method by which an intruder (Eve) could compromise this message exchange.

3 Eve can compromise this ~~each~~ exchange by providing an invalid public key to Alice. Thus, Alice will produce a cipher text that Bob cannot decrypt. (ends communication between Alice & Bob)
Eve can also steal Bob's private key, thus, allowing her to read Alice's messages to Bob.