Carleton University

School of Computer Science

Ⓑ

[redacted]

COMP 3000 (WINTER 2024)
OPERATING SYSTEMS --- FINAL EXAM --- VERSION B

INSTRUCTIONS

There are 26 questions on 8 pages worth a total of 35 points. You have 120 minutes. Please use the provided **bubble sheet** for the multi-choice/multi-select questions. For the short-answer questions, make sure to write your answers inside the bounding boxes (or the TA may miss part of your answer). Your answers should be concise and to the point (generally no more than a few sentences, sometimes much less). If you find a question to be ambiguous or believe it has issues, explain your interpretation and answer the question accordingly.

MULTI-CHOICE/SELECT QUESTIONS

1. [1] There are two types of pipes, i.e., named (FIFO) and unnamed (regular). With regard to their differences, which one of the following statements is INCORRECT?
   - A. Unnamed pipes can be created using the `pipe` command
   - B. Unnamed pipes can only be used by the process that created it and its forked processes
   - C. Named pipes have their corresponding files on the file system so that they can remain accessible even after the process that created it has finished execution (terminated).
   - [✓ D.] The FIFO file has its own FIFO inode type without data blocks
2. [1] Regarding the flags argument of the system call `mmap()`, which of the following achieves mapping a file into the calling process's address space such that changes made are reflected in the actual file on the storage device?
   - [✓ A.] `MAP_SHARED`
   - B. `MAP_FILE`
   - C. `MAP_PRIVATE`
   - D. `MAP_SHARED | MAP_ANONYMOUS`
3. [1] In a PTE (page table entry), there exist several bits to represent properties of the corresponding page. Which bit in the following is NOT found in the PTE?

Ⓑ

A. user/kernel
B. setuid
[✓ C.] present/absent
D. accessed
E. RW

4. [1] Which one of the following differs from the rest, in terms of **storage media type** (where data physically resides)?
   [✓ A.] Page Cache
   B. L2 Cache
   C. TLB (Translation Lookaside Buffer) Page Table Cache
   D. L3 Cache

5. [1] Which of the following involves **public/asymmetric-key cryptography**, as discussed in this course? Hint: asymmetric-key crypto involves a public key and a private key.
   [✓ A.] Recent ransomware
   B. `/usr/bin/login` authenticating a user trying to log in as student with a password, on our course VM
   C. Address space layout randomization (ASLR)
   D. Preventing student from writing to a file owned by root with `-rw-------`

6. [1] Match the three common authentication factors with their typical examples. Here the factors are denoted as: **K** = what you know (knowledge), **P** = what you have (possession), **I** = what you are (inherence)
   A. iris (**I**); pattern (for unlocking phones) (**K**); password (**P**); PIN (**K**)
   B. pattern (for unlocking phones) (**K**); voice (**I**); iris (**I**); password (**I**)
   [✓ C.] passphrase (**K**); RSA SecurID (**P**); USB token (**P**); fingerprint (**I**)
   D. face (**I**); passphrase (**I**); fingerprint (**I**); smart card (**P**)
   E. USB token (**I**); fingerprint (**I**); password (**K**); pattern (for unlocking phones) (**K**)

7. [1] Code in the ______ does NOT necessarily run in the kernel mode. Choose one of the options to fill in the blank, to make this statement correct.
   A. task scheduler
   [✓ B.] page fault handler
   C. device drivers
   D. SSHFS
   E. linux security modules (LSMs) K image

8. [1.5] Match the following operations with the main entity that performs them (as discussed in this course). Here the entities are denoted as: **K** = kernel-space code, **HW** = Hardware, **U** = User-space code
   A. Page fault handling**[K]**; Swapping out a page**[K]**; Page table walk**[HW]**; `malloc()` from existing heap**[U]**; Signal handling**[U]**; Running eBPF code**[K]**

tracing processes/calls

Ⓑ

B. Page fault handling**[HW]**; Swapping out an app**[K]**; Page table walk**[K]**; `malloc()` from existing heap**[U]**; Signal handling**[K]**; Running eBPF code**[HW]**

C. Page fault handling**[U]**; Swapping out an app**[K]**; Page table walk**[HW]**; `malloc()` from existing heap**[HW]**; Signal handling**[K]**; Running eBPF code**[HW]**

[✓ D.] Page fault handling**[U]**; Swapping out an app**[U]**; Page table walk**[HW]**; `malloc()` from existing heap**[K]**; Signal handling**[K]**; Running eBPF code**[U]**

9. [1.5] After you power on your computer with a regular Ubuntu installed, what should be the right order of the following steps? Assume all the steps are applicable to your computer.

   [✓ A.] BIOS/UEFI--> MBR ➞ /boot/grub/* ➞ /boot/vmlinuz ➞ /usr/sbin/init ➞ /usr/bin/login ➞ /usr/bin/bash

   B. MBR ➞ /boot/vmlinuz ➞ /usr/sbin/init ➞ /usr/bin/bash ➞ /boot/grub/* ➞ /usr/bin/login ➞ BIOS/UEFI

   C. /boot/vmlinuz ➞ /usr/sbin/init ➞ /usr/bin/bash ➞ BIOS/UEFI ➞ /usr/bin/login ➞ MBR ➞ /boot/grub/*

   D. MBR ➞ /boot/vmlinuz ➞ /boot/grub/* ➞ /usr/bin/login ➞ /usr/sbin/init ➞ /usr/bin/bash ➞ BIOS/UEFI

   E. /boot/grub/* ➞ /usr/bin/bash ➞ /boot/vmlinuz ➞ /usr/sbin/init ➞ /usr/bin/login ➞ BIOS/UEFI ➞ MBR

10. [1.5] When using a virtual address to locate a physical page, in what order usually do the following steps happen? Regular CPU caches are not considered here.

    [✓ A.] is the PTE in TLB? → page table walk → page fault → try loading page from disk → memory full and page something out

    B. is the PTE in TLB? → memory full and page something out → try loading page from disk → page table walk → page fault

    C. is the PTE in TLB? → page table walk → try loading page from disk → page fault → memory full and page something out

    D. try loading page from disk → is the PTE in TLB? → memory full and page something out → page fault → page table walk

    E. is the PTE in TLB? → page fault → page table walk → memory full and page something out → try loading page from disk

11. [1.5] Match the following memory paging design choices with their most likely major consequences as discussed in this course.

    A. too small pages ➞ external fragmentation; too big pages ➞ internal fragmentation; varying page sizes ➞ internal fragmentation; flat (1-level) page table ➞ page table taking too much space

Ⓑ

B. too small pages → page table taking too much space; too big pages → page table taking too much space; varying page sizes → external fragmentation; flat (1-level) page table → page table taking too much space

[✓ C.] too small pages → page table taking too much space; too big pages → internal fragmentation; varying page sizes → external fragmentation; flat (1-level) page table → page table taking too much space

D. too small pages → page table taking too much space; too big pages → internal fragmentation; varying page sizes → page table taking too much space; flat (1-level) page table → page table taking too much space

12. [1] Which one of the following can be a **possible solution** to the problem of deadlocks?

A. Lock
B. Condition variable
C. Semaphore
[✓ D.] Timeout
E. Mutex

13. [1] Considering **the OS topics we have discussed in this course**, select one from the listed terms that does NOT pertain to the same topic as the rest of the terms do.

[✓ A.] page fault mem
B. dentry files
C. mount
D. logical size files
E. superblock ?

14. [1] Select one statement that is correct.

A. As its name implies, rootkit's privilege should be higher than or at least equal to that of the root user
[✓ B.] Key loggers can take the form of either software or hardware
C. The adversary is unable to extract the plaintext passwords from `/etc/shadow` because they do not have the key (which is always kept secret)
D. Library/function calls jump to user-space addresses, system calls jump from a user-space address to a kernel-space address

15. [1] Based on what was discussed in this course, which one of the following is NOT a building block for containers (containerization)?

A. namespaces
B. capabilities
C. control groups
[✓ D.] mandatory access control

Ⓑ

16. [1] Which are the restrictions applied to eBPF code making it **safer** compared to directly running code in kernel modules or the kernel image? Choose three (No partial marks).
    [✓ A.] no default access to kernel functions
    B. memory access bounds checked
    [✓ C.] being JIT compiled
    D. safety-checked before loaded
    [✓ E.] loaded by a frontend like `bpftrace`

17. [1] For a security solution implemented in a **user-space application**, what is considered part of its TCB? Choose the best (most complete) one.
    A. The hypervisor
    B. Other user-space applications
    C. The hypervisor, OS, and UEFI/BIOS
    D. The OS kernel
    [✓ E.] Other user-space applications, the hypervisor, OS, and UEFI/BIOS

18. [1] Which one of the following statements about containers is INCORRECT?
    [✓ A.] Linux Security Modules (LSMs) must be used to construct a container
    B. Sharing the same OS kernel can be both an advantage and a disadvantage for containers, compared to VMs
    C. Usually, a container management tool can make use of different execution engines, and a container execution engine may also be used by various container management tools
    D. Docker features single-application containers and a layered file system for image file reuse/sharing

19. [1] With respect to building/using a Linux kernel module, which statement is correct?
    A. As with shared libraries (`.so`) in user-space, kernel modules (`.ko`) also have the purpose of avoiding multiple (redundant) copies of the same code in RAM.
    B. The tool `dkms` will be needed if you want to build your kernel module ✓
    C. The available functions that can be called in a kernel module are very likely the same as the functions you can use in a user-space program
    [✓ D.] As kernel modules are very specific to kernel versions, you will not be able to load a kernel module built for a different (mismatched) kernel version without forcing it ✓

20. [1] When does kernel module code normally run? Choose the one that is **incorrect** (or very unlikely).
    A. When `/dev/zero` is being read
    [✓ B.] Continuously in the background
    C. In response to registered events
    D. When a module is unloaded

E. When a module is loaded

Ⓑ

SHORT-ANSWER QUESTIONS

21. [2] Based on what we have learned in the memory management topic, what the OS can do when the system is running low on memory space (RAM)? Hint: two options. If you don't remember the exact terms, you can describe the operations in your own words.

[redacted]

22. [2] To determine which page in RAM to swap out to disk, we have discussed a simple algorithm called the "**clock algorithm**" considering the fact that we can only use one bit in the PTE. Briefly explain how the clock algorithm works to achieve its purpose.

[redacted]

Ⓑ

23. [3] Assume that you are the sysadmin of a computer and would like to configure the web server program `/usr/local/apache2` for user **student**. The server program needs to listen on port 80 and student is NOT listed in `/etc/sudoers`. Which **two configuration options** do you have so that **student** can start the program every day to listen on port 80? What is an obvious **advantage** of one option over the other? Be specific and clear (only mentioning a few keywords will not suffice for full marks).

[redacted]

24. [2] Which one of the following involves **cryptography** (as opposed to just **access control**)? Explain why briefly (only for the selected one).

A. Determining whether "student" can access a file "tut8"

B. Crashing a program of user "student" attempting to access a kernel object

[✓ C.] Starting a process (address space) with ASLR enabled

D. Preventing "student" from writing to a file owned by root with permission bits 0600

[redacted]

Ⓑ

25. [2] What are the **four necessary conditions** for a deadlock to happen? Be specific. You can use examples if you find them useful.

[redacted]

26. [2] With regard to the performance of IPC (Inter-Process Communication) mechanisms, is messaging (e.g., pipes or sockets) usually faster or slower than shared memory? Mention **two factors** that contribute to this.

[redacted]