

Name:

ID Number (last 4 digits):

Exam Type:

Exam Number:

Grade: 1: ... 2: ... 3: ... 4: ... 5: ... 6: ... 7: ... Total:

THERE ARE 4 (FOUR) PAGES THIS PAGE INCLUDED IN THIS EXAMPAPER

Solve all the problems in 70min in the space provided in this exampaper

Read the Instructions and Problems carefully; do not complain later

Make sure you sign the exam before you turn it in
Unsigned exams are marked with grade 0 (zero).

Signature

.....

List of useful formulae

NOTE: This exam is open (paper-version) textbook. Only pens/pencils and this exampaper are allowed on the table. For scratch paper, use the rear side of an exam page marking it as needed.

WARNING: MOBILE PHONES OR DEVICES PER HANDOUT 1 (SYLLABUS) MUST BE POWERED DOWN AND IN BACKPACK/BAG. OTHERWISE SCREEN FACING THE FLOOR AND ON THE FLOOR.

Problem 1. (50 POINTS)

TRUE or FALSE. Circle one and only one. T or F marks are ignored.

1. TRUE FALSE. A primary objective of an OS is to manage hardware resources and provide services to users.
2. TRUE FALSE. Two different processes in Linux can have the same PID (process ID).
3. TRUE FALSE. Process management code in the kernel is always executed in user-mode.
4. TRUE FALSE. A (single) fork() in Linux can create two processes.
5. TRUE FALSE. Context switching in multi-tasking (eg CTSS) is as expensive as context switching in multi-programming.
6. TRUE FALSE. A user-mode process can modify the PCB of its own process.
7. TRUE FALSE. It is possible for a process to have its state changed from SUSPEND directly into RUNNING.
8. TRUE FALSE. The best way to exploit temporal locality is to store recently used instructions or data in the cache.
9. TRUE FALSE. I/O instructions in a user-program are executed in kernel-mode.
10. TRUE FALSE. System calls in a user-program are executed in user-mode.

Problem 2. (50 POINTS)

Answer briefly or as specified the following questions. The shorter the correct answer is the more of the points you will receive.

1. Give a definition of the term process in no more than five words.
2. Programs and data are stored in different memories. What architecture is it? Circle the answer.

Harvard von-Neumann.
3. A UNIX/Linux fork that succeeds what positive value does it return to the parent process?
4. What are the two types of locality (of reference)?
5. Why is locality (of reference) important?
6. When a system call occurs in a user program, a kernel stack is used that is separate from the user stack. Why?
7. What is the difference between Blocked/Suspend and Blocked state for a process?
8. Name three differences between the process image of the parent process and the process image of a child process successfully created by that parent at the completion of the fork() operation.
9. Which one of the numbers is more typical to be an L2 cache size of a general purpose CPU nowadays? Circle the correct answer.

16, 1,024B , 256KiB , 32KiB, 16MiB.
10. An I/O device issues a **trap** or an **interrupt**? Circle the correct bold-faced word.

Problem 3. (30 POINTS)

Fill the following table for the cache hierarchy described, using the definition terminology used in class and the notes. A 20MiB L3 cache is 20-way associative with 64B block size. It is part of a 64-bit architecture but only the lower 48 bits are being used for addressing; the remaining bit are ignored.

(3.1) A (associativity) =
 (3.2) C (line size) =
 (3.3) K (block size) =
 (3.4) C num of bit =
 (3.5) K num of bit =
 (3.6) T num of bit (tag)=

Problem 4. (30 POINTS)

Consider a memory system with a cache and main memory. It takes 10ns to access the cache, and 100ns to access the main memory. If the effective (memory) access time is 200% of the cache access time, what is the hit ratio H ?

Problem 5. (30 POINTS)

You execute the following program. Assume that the fork succeeds in executing. What is the value printed by the program below (i.e. the value of fval)? Let the parent process have PID 2345 and the child process created has PID 4567.

```
#include <stdio.h>
#include <sys/types.h>
int main(void) {
    int fval;
    fval=fork();
    if (fval > 1000) {
        printf("%d\n", fval); /* Print the value of fval */
    }
    return(0);
}
```

Problem 6. (30 POINTS)

A CS332 year is 12 months, each month has 30 days, and each day 24 hours. A data center contains 1,000 identical HDDs. A HDD's MTTF was measured based on the characteristics of those HDDs of the data center and was found to be 1,440,000h. How many HDD failures do you expect over a CS332 year?

Problem 7. (30 POINTS)

We have two jobs A , B . The execution pattern for A is CPU first I/O next, and of B I/O first CPU next. Each one needs CPU time of 20min. The I/O time of A is $2/3$ of the total execution time of A . Likewise for B . Ignore context switch time or initial program load time (assume that it is zero).

($3/5$ is a minimal integer fraction, $6/10$ is not a minimal integer fraction even if it is an integer fraction, $1.5/2.5$ is NOT an integer fraction, and 60% means the same thing as the other three.)

(a) **Serial execution.** What is the elapsed time (i.e. turnaround time) of A , B ? Both A, B are available at the same time but A is first. What is the CPU utilization after the completion of the last job? (Express it as an integer percentage rounded down or minimal integer fraction.)

(b) **Multiprogramming (no preemption).** Assuming that both arrive at the same time, what is the elapsed time of A , B ? What is the CPU utilization after the completion of the last job?

(c) **Multitasking (timesharing) with slice of 10min: a program has control overall resources during the slice. Multiprogramming is not available.** Assuming that both arrive at the same time but A has a slightly faster arrival. What is the elapsed time of A , B ? CPU utilization ?