

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 the exam.

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. (40 POINTS)

TRUE or FALSE. Circle one and only one. Any marks with a T or F anywhere else would be considered wrong.

1. **TRUE** **FALSE**. A primary objective of an OS is to manage hardware resources and provide a useful interface of those resources to programmers.
2. **TRUE** **FALSE**. Every process in Linux has its own unique PID (process ID).
3. **TRUE** **FALSE**. Process management code in the kernel is always executed in kernel-mode.
4. **TRUE** **FALSE**. Shortest process next is a non preemptive policy.
5. **TRUE** **FALSE**. Switching between threads is as costly in time as switching between processes.
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 BLOCKED directly into RUNNING.
8. **TRUE** **FALSE**. A process terminating also terminates all the threads created by the process itself.

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. **Harvard** **von-Neumann**. Programs and data are stored in the same one memory. What architecture is it? Circle the answer.
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. Below the seven states of the seven state process model are numbered 1-7. Give the number-index of the state(s) for which a process is in main memory. For example one answer is 6 as a Running process must be in main memory. Exclude from this answer also states 1 and 7 and thus list only states 2-6, as needed/required.
 - (a) New
 - (b) Read/Suspend
 - (c) Blocked/Suspend
 - (d) Ready
 - (e) Blocked
 - (f) Running
 - (g) Exit
9. Which of the four numbers is more typical to be the L1D cache size of a general purpose CPU nowadays. Circle the correct answer:

16, 1,024B , 32KiB, 32MiB.
10. What entity is called 'a lightweight process'?

Problem 3. (30 POINTS)

Fill the following table for the cache hierarchy described, using the definition terminology used in class and the notes. A 1MiB L2 cache is 8-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.

(2.1)	A	(associativity)	=
(2.2)	C	(line size)	=
(2.3)	K	(block size)	=
(2.4)	C	num of bit	=
(2.5)	K	num of bit	=
(2.6)	T	num of bit	=

Problem 4. (30 POINTS)

Consider a memory system with a cache and main memory. It takes 20ns to access the cache, and 200ns to access the main memory. If the effective (memory) access time is 15% of the main memory access time, what is the hit ratio H ?

Problem 5. (30 POINTS)

You execute the following program. Assume all forks succeed in executing. How many hello will be output?

```
#include <stdio.h>
#include <sys/types.h>
int main(void) {
    printf("hello\n");
    fork();
    fork();
    fork();
    fork();
    return(0);
}
```

Problem 6. (30 POINTS)

Multiple jobs can run serially as described in class. Suppose that two jobs *A* and *B*, each needing 30 minutes of CPU time, start simultaneously at time 0. How long will the last one take to complete (i.e. what is its turnaround time) if they run serially? Assume that 50% of total execution time of a job is I/O wait and the pattern is CPU-I/O for *A* and I/O-CPU for *B*. What is the overall processor utilization (use a percentage) over the period of A,B's execution? (Be very very careful.)

Then we upgrade the computer system to allow multiprogramming. How long will the last job take to complete (i.e. what is its turnaround time) if they run in such a multiprogramming mode optimally? What is the overall processor utilization (use a percentage) over the period of A,B's execution?

Problem 7. (40 POINTS)

Five processes A, B, C, D, E arrive at almost the same time in the ready queue. They have estimated running times of 10,8,6,4,2 seconds respectively. Determine the mean (average) process turnaround time for each of the three cases listed. Ignore process (context) switching overhead. Assume that all processes are CPU-bound.

In lieu of a calculator: $1/5 = 0.2$, $2/5 = 0.4$, $3/5 = 0.6$, $4/5=0.8$.

- First-Come-First-Served (run in order C, B, D, A, E to break arrival tie).
- Shortest Process Next.
- Round-Robin with time slice of 2s and the initial order is A,B,C,D,E.