

CS 3113 Introduction to Operating System - Fall 2025 – Exam 1

Part 2 (Open Questions)

Tuesday, September 30, 2025

Exam Instructions

- The exam is **closed to electronics** but open to unlimited handwritten notes.
- Printed slides may be used **only if each page contains handwritten notes**. Pages without handwritten notes are not permitted.
- You may use a scientific calculator; however, **mobile phone calculators are not allowed**.
- You will have **75 minutes** to complete the exam (Part 1 and Part 2).
- Be sure to write your **full name** (exactly as shown on the Canvas roster) and your **OU ID** on your exam.

All the best!

13:20
14:45

Name:

Chik-Yu Chu

OU ID:

113595150

Total Score

Part 1-Multiple Choice Answer Sheet

Record your answers to the multiple-choice questions in the table below. Responses must be indicated using capital letters (A, B, C, or D) only. Each multiple-choice question is worth 1.5 point.

The complete set of Part 1's questions is provided on the separate sheets.

Questions #	Answer	Questions #	Answer
1.	B	11.	B
2.	C	12.	B
3.	B	13.	B
4.	B	14.	D
5.	A	15.	D
6.	D	16.	C
7.	B	17.	B
8.	B	18.	B
9.	B	19.	A
10.	C	20.	B

Part 2-Open Questions: Questions # 21 - 25

(Write your answers only in the space provided)

21. [15 points] Consider the following program, which uses fork() to create a child process. The program modifies and prints the elements of an array defined as a global variable in both the child and parent processes.

```
1 #include <sys/types.h>
2 #include <stdio.h>
3 #include <unistd.h>
4 #define SIZE 5
5     index 0 1 2 3 4
6 int data[SIZE] = {1, 2, 3, 4, 5};
7
8 int main()
9 {
10     int i;
11     pid_t q;
12
13     q = fork();
14
15     if (q == 0) {
16         for (i = 0; i < SIZE; i++) {
17             data[i] += 10;
18             printf("%d ", data[i]); /* Line 18 */ [11, 12, 13, 14, 15]
19         }
20     }
21     else if (q > 0) {
22         wait(NULL);
23         for (i = 0; i < SIZE; i++) {
24             printf("%d ", data[i]); /* Line 24 */
25         }
26     }
27
28     return 0;
29 }
```

- a. What will be the output at lines 18 and 24? (8 points)
b. Explain the behavior of the array values in both the child and parent processes. (7 points)

Answer Question 21 a:

Line 18: 11, 12, 13, 14, 15

Line 24:

1, 2, 3, 4, 5

Answer Question 21 b:

both have separate memory, both begin with $\{1, 2, 3, 4, 5\}$
for children, each number add 10, so become $\{11, 12, 13, 14, 15\}$
for parent, no change, so it's $\{1, 2, 3, 4, 5\}$

22. [10 points] What is a process and a thread? What is the main difference between multiprocessing and multithreaded application?

Answer Question 22:

Process : has its own entire memory space (code, global data, ...)

Thread : shares the same memory space as other threads within the same process, except for a few things like the thread stack and registers

process is also an active entity, programme in execution, including its state, resources, and program counter. it's an unit of work within the system

Multiprocess is multiple processor in a single computer sharing.

I/O & memory, general parallel processing within one system

Multithread also shared memory, global variable and code. Threads execute concurrently within the same process.

23. [20 points] Consider the following C code fragment

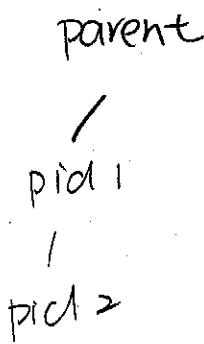
```
1 int main(int argc, char *argv[])
2 {
3     int a = fork(); < parent
4     if (a < 0) { children
5         fprintf(stderr, "fork failed\n");
6         exit(1);
7     } else if (a == 0) {
8         printf("%d\n", getpid());
9         int b = fork(); < parent
10        children
11        printf("%d\n", b);
12        printf("%d\n", getppid());
13        execl("/bin/ls", "ls", "-l", (char *)NULL);
14        int c = fork();
15        wait(NULL);
16        printf("%d\n", a);
17        printf("%d\n", getpid());
18        int d = fork();
19    }
20    return 0;
21 }
```

a=100
b=101
c=102
d=103

a) Assuming no errors when the program is run and `execl` on line 12 is successful, construct a process tree based on the code above. (10 points)

b) Suppose the process IDs for main, a, b, c, and d are 100, 101, 102, 103, and 104 respectively. What is the output of the print statements on lines 8, 10, 11, 16, and 17? (10 points)

Answer Question 23 a:



Answer Question 23 b:

- Line 8: 0
Line 10: 102
Line 11: 102
Line 16: 101
Line 17: 104

$$0.4 \times 25 = 10^4$$

24. [15 points] Suppose that 40% of a program's execution time must be run sequentially, while the remaining 60% can be parallelized. Using Amdahl's Law:

a) Calculate the maximum theoretical speedup achievable with 5 cores. Show your work. (8 points)

b) Determine the maximum theoretical speedup achievable with an infinite number of processors. Show your work. (7 points)

Answer Question 24 a:

$$\begin{cases} 40\% \text{ sequentially } \rightarrow S = 0.4 \\ 60\% \text{ parallelized. } \rightarrow P = 0.6 \\ \text{core} = 5 \quad \rightarrow N = 5 \end{cases}$$

$$\begin{aligned} S(N) &= S(5) = \frac{1}{0.4 + \frac{0.6}{5}} \\ &= \frac{1}{0.52} = \frac{100}{52} \\ &= 1.923 = 1.92 \cancel{\text{xx}} \end{aligned}$$

Answer Question 24 b:

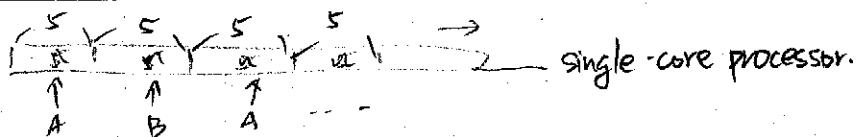
$$\begin{cases} S = 0.4 \\ P = 0.6 \\ N \rightarrow \infty \end{cases} \quad S(N) = \frac{1}{S + \frac{P}{N}} = \frac{1}{0.4 + \frac{0.6}{\infty}} = \frac{1}{0.4} = 2.5$$

$$S(N) = \frac{1}{S} = \frac{1}{0.4} = \frac{10}{4} = 2.5$$

The maximum theoretical speedup is 2.5 *

25. [10 points] Imagine two processes, A and B, are running on a single-core processor, managed by the kernel using a time-sharing method with a time quantum of 5 units. Clearly explain what a context switch means in this situation. Describe what happens during a context switch.

Answer Question 25:



- Context switch means switching from one process to another while switching.
- Context switch doesn't perform useful work, so there is no work during switching.
- In this case, time-sharing method with a time quantum of 5 units, mean that before process termination (namely before either process A or process B finish), the kernel changes the process for every 5 units. Only one process is working on kernel at one time, and after 5 units, no matter it finish or, there will be context switch.