CS 3113 Introduction to Operating Systems
Final Exam
December 12, 2018

General instructions:

- Please wait to open this exam booklet until you are told to do so.
- This examination booklet has 16 pages. You also have been issued a bubble sheet.
- Write your name, university ID number and date, and sign your name below. Also, write your name and ID number on your bubble sheet, and fill in the bubbles for your ID.
- The exam is closed book, notes and electronic devices. The exception is that you may have one page of personal notes (double sided).
- The exam is worth a total of 137 points (and 15% of your final grade).
- You have 2 hours to complete the exam. Be a smart test taker: if you get stuck on one problem go on to the next.
- Use your bubble sheet to answer all multiple-choice questions. Make sure that the question number and the bubble row number match when you are answering each question. If you cannot effectively erase an incorrect answer, mark an 'X' over it.

On my honor, I affirm that I have neither given nor received inappropriate aid in the completion of this exam.

Signature: _____

Name: _____

ID Number: _____

Date: _____

Question	Points	Score
Files and File Systems	24	
Memory Management	33	
Program Compilation and Makefiles	7	
System Calls	10	
Processes and Threads	22	
Characters and Strings	12	
Resource Sharing	29	
Total:	137	

Part I. Files and File Systems

1. (2 points) True or False? *read()* alters the offset stored in the *Open File Table*.
   A. True B. False

2. (4 points) Assume the initial contents of file *myfile2* and the execution of the following block of code. What are the final contents of myfile2? Assume that there are no errors.

   myfile2 initial contents:

```
foobarbaz
```

   Code block:

```
char buf[100];
int fd = open("myfile2", O_RDWR);
lseek(fd, 3, SEEK_SET);
read(fd, buf, 3);
lseek(fd, 0, SEEK_SET);
write(fd, buf, 2);
close(fd);
```

   A. bafoobarbaz **B. baobarbaz** C. barbarbaz D. foobarbaz
   E. Answer not shown

3. (2 points) True or False? When two processes simultaneously and independently open the same file name, they share the file offset.
   A. True **B. False**

4. (3 points) Assume that a disk block stores N inodes and that there are multiple, sequential blocks that store inodes. Assume also that the first inode block is disk block M. Given that we would like to access inode i, which block and element of the block, respectively, must we access?
   A. $N + i\%M$ and i/M
   B. $N + i/M$ and $i\%M$
   C. $M + i\%N$ and i/N
   D. $M + i/N$ **and** $i\%N$
   E. Answer not shown

5. (4 points) Assume the initial contents of file *myfile* and the execution of the following block of code. What are the final contents of myfile? Assume that there are no errors with permissions and that there are no newlines.

   myfile initial contents:

```
pass
```

   Code block:

```
FILE *fp = fopen("myfile", "w");

if(fp == NULL){
  printf("Error");
  exit(-1);
}

fprintf(fp, "_the exam");

fclose(fp);
```

   A. pass **B. _the exam** C. pass_the exam D. There is an error
   E. Answer not shown

6. (4 points) Assume that a directory is deleted from the OUFS; the directory's inode index is 22. Before the deletion, the first three entries of the inode allocation table are: $0xFF$, $0x4C$ and $0xD7$. After a successful deletion, what is the new state of the inode allocation table (first 3 entries)?

   A. $0xFF$, $0x4C$, $0x97$ B. $0xFF$, $0x0C$, $0xD7$ C. $0xFF$, $0x4C$, $0xD3$
   D. $0xFF$, $0x4C$, $0x57$ E. Answer not shown

7. (2 points) True or False? With respect to the data storage for a file in OUFS, the system is subject to internal fragmentation.

   A. True B. False

8. (3 points) Assuming no errors and that a pipe is still open, a *read()* on that pipe for **N** bytes will block until which of the following:

   A. At least one byte is available to return
   B. N bytes are available to return
   C. *read()* does not block
   D. Answer not shown

Part II. Memory Management

9. (4 points) Assume a fixed partitioning scheme and that the *buddy system* is being used to allocate space. The physical memory is composed of 16 frames (0...F). The current state of allocation is indicated below, with process number or 'x' indicating the allocation state of the small blocks (x=unallocated). A new process is brought into main memory that requires two frames. Which frames are allocated? Assume that the lowest numbered valid frames are chosen.

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
6	x	x	1	2	2	4	x	x	x	x	5	3	3	3	3

A. 1 and 2 B. 7 and 8 **C. 8 and 9** D. 9 and A E. Answer not shown

10. (3 points) Consider a hybrid segmentation / paging memory management scheme, with page sizes of 2^5 and page table sizes of 2^4.

What is the maximum memory allocation possible for a segment?

A. 2^4 bytes B. 2^5 bytes C. $2^4 + 2^5$ bytes **D. 2^9 bytes**
E. Answer not shown

11. (4 points) With reference to the above paged segmentation scheme, three segments have been allocated to the currently executing process. Their page tables are shown below (only valid table entries are shown):

S0, Len: $0x47$
$0x4$
$0xA$
$0x7$

S1, Len: $0x9A$
$0x2$
$0xC$
$0xD$
$0x1$
$0xF$

S2, Len: $0x15$
$0xB$

Given a logical address of $0x287$, which physical address is accessed?

A. $0xF7$ B. $0x147$ C. $0x187$ **D. $0x1E7$** E. Answer not shown

12. (3 points) Below is a list of properties of memory (assuming a C compiler). Mark the one item that is more true for the stack than the heap.

A. Slower allocation
B. Size determined at run time
C. Local memory storage
D. Global memory storage
E. Persistent after function returns

13. (4 points) Consider a simple segmented memory system, with a maximum segment size of 2^{10} and a segment table as follows:

Length	**Base**
$0x10$	$0x280$
$0xA3$	$0x300$
$0x40$	$0x220$
$0x200$	$0x0$

Given a logical address of $0x11$, which physical address is accessed?

A. $0x281$ B. $0x291$ C. $0x2A1$ D. $0xA0011$ **E. Answer not shown**

14. (4 points) With reference to the above segmented memory scheme, given a logical address of $0x835$, which physical address is accessed?

A. $0x235$ **B.** $0x255$ C. $0x335$ D. $0x8B5$ E. Answer not shown

15. (2 points) True or False? In a virtual memory management scheme, for a process to be in a running state, all of its pages must be assigned to physical memory.

A. True **B. False**

16. (2 points) True or False? In a paging memory management scheme, one way to reduce internal fragmentation is to increase the page size.

A. True **B. False**

17. (4 points) Consider a simple paged memory system, with page sizes of 2^6 and a page table as follows:

$0x5$
$0x7$
$0xC$
$0x2$

Given a logical address of $0xB6$, which physical address is accessed?

A. $0x36$ B. $0x1F6$ **C.** $0x336$ D. $0xC36$ E. Answer not shown

18. (3 points) In the Linux EXT file system, access permission information is stored where?

A. The directory entry **B. The file's inode** C. The process control block
D. The master block E. Answer not shown

Part III. Program Compilation and Makefiles

19. (3 points) The *gcc* executable performs a number of different functions. Which one of these is responsible for creating an object file from a C file?

A. Linker B. C Preprocessor **C. Compiler** D. Answer not shown

20. (4 points) Consider a program that is implemented as two distinct source files (part1.c and part2.c) to create a single executable *zformat*. On which line is the bug in the following Makefile that prevents the executable from being compiled?

```
1  CC = gcc
2  CFLAGS = -Wall -g
3  ALL = part1.o part2.o
4
5  all: $(ALL)
6
7  part1.o: part1.c
8          $(CC) -c $(CFLAGS) part1.c -o part1.o
9
10 part2.o: part2.c
11         $(CC) -c $(CFLAGS) part2.c -o part2.o
12
13 zformat: part1.o part2.o
14         $(CC) part1.o part2.o -o zformat
15
16 clean:
17      rm *.o $(ALL)
```

A. 3 B. 7 C. 13 D. 14 E. Answer not shown

Part IV. System Calls

21. (2 points) True or False? If a process does not share a file with other processes, then it can access that file without the use of system calls.

    A. True **B. False**

22. (2 points) True or False? *qsort()* is a system call.

    A. True **B. False**

23. (2 points) True or False? A process can allocate a new stack frame in user mode.

    A. True B. False

24. (2 points) True or False? A system call results in the processor entering into kernel mode.

    A. True B. False

25. (2 points) True or False? A user program can add new **sys calls** to improve program efficiency.

    A. True **B. False**

Part V. Processes and Threads

26. (3 points) Which of the following statements is false?

A. Process copy on write can decreases process creation time
B. Thread execution allows each processor to be used
C. Separate threads are executed in isolated memory spaces
D. Multiple spawned threads from the same process share a process ID

27. (2 points) True or False? When using user-level threads, the creation of a new thread involves a system call.

A. True **B. False**

28. (2 points) True or False? When measuring execution time for a user process, in a multi-threaded application and multi-core environment, the *user_time* will reflect the total wall clock time.

A. True **B. False**

29. (2 points) True or False? A program can be composed of multiple processes, each with multiple threads.

A. True B. False

30. (3 points) Which of the following is a method of combining I/O calls to reduce blocking time?

A. Encapsulation
B. Jacketing
C. Kernel blocking
D. User-level blocking
E. Answer not shown

31. (3 points) Which of the following is an attribute that **is not included** as part of POSIX threads or Linux processes

A. An identifier B. A priority C. A status value **D. A boot block**
E. Answer not shown

32. (4 points) How many stars are output by this program?

```
int main(int argc, char** argv)
{
  for(int i = 1; i < 4; ++i) {
    int pid = fork();
    if(pid == 0) {
      i++;
      write(1, "*", 1);
    }else{
      wait(NULL);
    }
  }
}
```

A. 0 B. 3 **C. 4** D. 9 E. Answer not shown

33. (3 points) Which of the following is the worst use case for threads?

A. Managing foreground and background GUI work
B. Periodic backup for an user application
C. Handling incoming I/O
D. Handling interaction with multiple, independent users

Part VI. Characters and Strings

34. (4 points) What is printed by the following block of code? (do not worry about spaces and newlines in your answer)

```
char str[100];

strcpy(str, "IBM");

int len = strlen(str);

for(int i = 0; i < len; ++i)
    str[i]--;

printf("%s\n", str);
```

A. HAL B. ibm C. IBM D. JCN E. Answer not shown

35. (4 points) What is output by the following block of code? (do not worry about spaces and newlines in your answer)

```
char str[100];

strcpy(str, "Stone and sea are deep in life");

int len = strlen(str);

for(int i = 0; i < len; ++i)
   if(str[i] == 'a')
      str[i] = 0;

printf("%s\n", str);
```

A. Stone
B. Stone nd se re deep in life
C. Stone and sea are deep in life
D. Stone And seA Are deep in life
E. Answer not shown

36. (4 points) What is output by the following block of code? (do not worry about spaces and newlines in your answer)

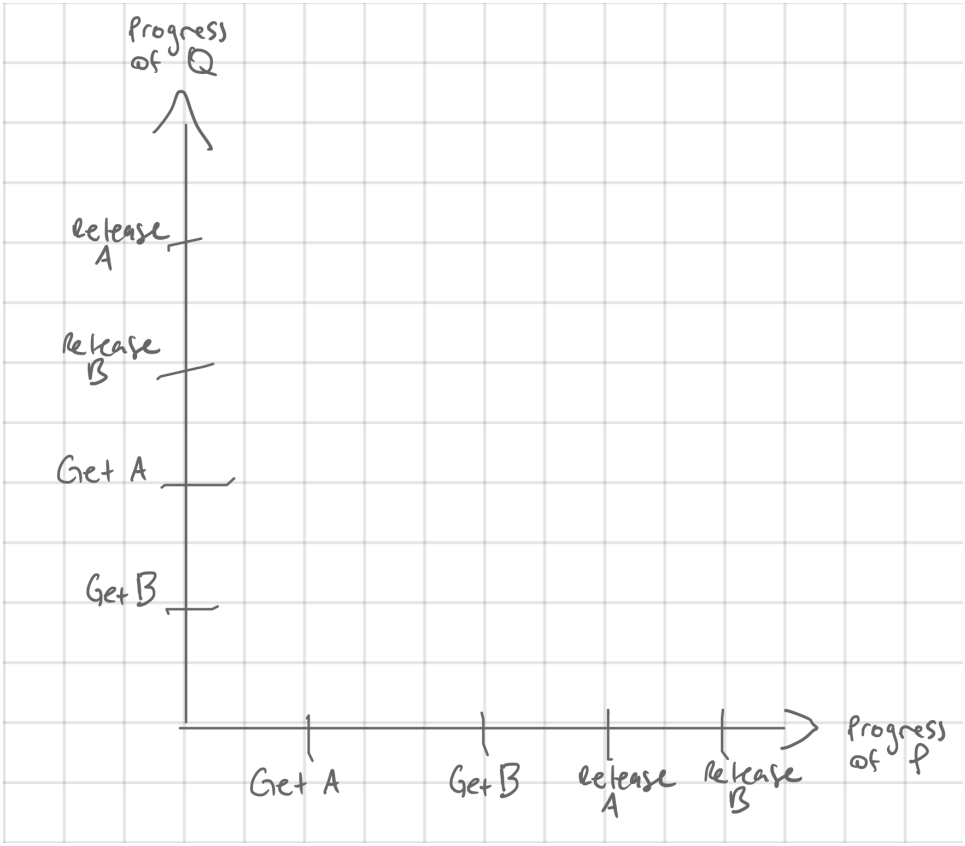
```
char str[100];

strcpy(str, "Permanence at rest and permanence in motion");

printf("%s\n", &(str[19]));
```

A. a
B. and
C. and permanence in motion
D. Permanence at rest and permanence in motion
E. Answer not shown

Part VII. Resource Sharing

37. (4 points) Given the incomplete *joint process diagram*, which order of process execution will result in inevitable deadlock? Assume that exactly one unit each of A and B are available.

A Joint Process Diagram



A. Q acquires B and then A and then releases B and A
B. Q acquires B and then P acquires A
C. P acquires A and then B and then releases A and B
D. Answer not shown

38. (3 points) A semaphore whose definition includes the policy that the process that has been blocked the longest is released from the queue first is called which of the following?

A. a general semaphore
B. a strong semaphore
C. a weak semaphore
D. a counting semaphore

39. (3 points) A situation in which a runnable process is overlooked indefinitely by the scheduler, although it is able to proceed, is which of the following?

A. mutual exclusion
B. deadlock
C. starvation
D. livelock

40. (4 points) According to the Banker's algorithm, which process(es) cannot be guaranteed to be fulfilled.

Claim Matrix

	Camera	**Printer**	**Bluetooth**
Process A	0	3	2
Process B	2	1	4
Process C	3	1	1

Current Allocation

	Camera	**Printer**	**Bluetooth**
Process A	0	2	1
Process B	1	0	2
Process C	1	1	0

Max Resources

Camera	**Printer**	**Bluetooth**
4	3	3

A. Process A B. Process B C. Process C D. Process B & C
E. Answer not shown

41. (3 points) Which condition of deadlock is prevented by requiring that resource types are allocated in a specific linear order?

A. hold and wait B. no preemption C. exclusion **D. circular wait**

42. (4 points) True or False? According to the Banker's algorithm, the following system state is safe.

Claim Matrix

	Camera	**Printer**	**Bluetooth**
Process A	0	3	2
Process B	2	1	4
Process C	3	1	1

Current Allocation

	Camera	**Printer**	**Bluetooth**
Process A	0	2	1
Process B	1	0	2
Process C	1	1	0

Max Resources

Camera	**Printer**	**Bluetooth**
5	3	4

A. True B. False

43. (4 points) Below are two example processes that are using a shared variable *turn* to manage progress. Which of the following statements are true?

```
/*PROCESS 0 */
while (turn != 0)
  /* do nothing */;

/*critical section */;
turn = 1;
```

```
/*PROCESS 1 */
while (turn != 1)
  /* do nothing */;

/*critical section */;
turn = 0;
```

A. If process 0 fails in the critical section; process 1 cannot continue
B. The turn variable is equivalent to a binary semaphore
C. The process 0 speed of execution can skew the progress between the processes
D. All the above are true
E. None of the above are true

44. (4 points) Consider the following solution to the dining philosophers problem where the philosophers are seated at a round table with forks to their left and right (which are shared with their neighbors). Can deadlock happen? Assume that *token* is unique to the table, but that *fork_left* and *fork_right* refer to the left/right fork for each philosopher.

```
while(true){
    think()
    wait(token)
    wait(fork_left)
    wait(fork_right)
    signal(token)
    eat()
    signal(fork_right)
    signal(fork_left)
}
```

A. Yes **B. No**