CS 3113 Introduction to Operating Systems
Midterm
October 11, 2018

General instructions:

- Please wait to open this exam booklet until you are told to do so.
- This examination booklet has 10 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 1.25 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	39	
Memory and Memory Management	21	
Program Compilation	13	
System Calls	15	
Processes	37	
Serial Processing	12	
Total:	137	

Part I. Files and File Systems

1. (3 points) True or False? When two processes simultaneously and independently open the same file name, they maintain independent file contents.

   A. True **B. False**

2. (3 points) True or False? The read() system call always returns the number of bytes read.

   A. True **B. False**

3. (3 points) True or False? In the Linux ext file system, an i-node contains the contents of a file.

   A. True **B. False**

4. (6 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 no errors with permissions.

   myfile2 initial contents:

```
foo bar baz
```

   Code block:

```
int fd = open("myfile2", O_WRONLY);

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

if(lseek(fd, 3, SEEK_SET) == -1) {
  printf("Error");
  exit(-1);
}

write(fd, "BAZ", 3);

close(fd);
```

   A. BAZ B. foo bar baz C. foo bar baz BAZ D. foo BAZ baz
   E. Answer not shown

5. (6 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 no errors with permissions and that there are no newlines.

   myfile initial contents:

```
pass
```

   Code block:

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

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. (3 points) True or False? When two processes simultaneously and independently open the same file name, they maintain independent file offsets.

   A. True B. False

7. (6 points) Given the following partition configuration and the commands below, what is printed by the shell from the last command? (ignore spacing and newlines in the answer)

   Partition: /dev/sda1, mounted as root (i.e., /)

```
/
   tmp/
   bin/
   scratch/
      data/
   home/
      linus/
      bob/
```

   Partition: /dev/sdb2

```
/
   brian/
      projects/
      notes/
   ada/
      difference/
      engine/
```

   Shell commands:

```
   mount -t ext3 /dev/sdb2 /home
   ls /scratch
```

   A. data B. brian ada C. linus bob D. File not found
   E. Answer not shown

8. (6 points) Given the above configuration and mount command, what is printed by the shell in response to this command? (ignore spacing and newlines in the answer)

```
   ls /home
```

   A. data **B. brian ada** C. linus bob D. File not found
   E. Answer not shown

9. (3 points) True or False? In the Linux ext file system, an i-node contains information about the owner of a file.

   A. True B. False

Part II. Memory and Memory Management

10. (5 points) As you travel up the memory hierarchy (to the smallest-sized memory) which of the following is true?

    A. Decreasing cost per bit and increasing access time
    B. Increasing cost per bit and increasing access time
    C. Increasing cost per bit and decreasing access time
    D. Decreasing cost per bit and decreasing access time
    E. Answer not shown

11. (5 points) Below is a list of properties of memory (assuming a C compiler). Mark the one item that is more true for the heap than the stack.

    A. Faster allocation
    B. Automatic memory management
    C. Local memory storage
    D. Global memory storage
    E. Persistent after function returns

12. (5 points) A new stack frame is created when function **A** calls function **B**. This stack frame **does not** contain which one of the following?

    A. A copy of the registers used by function **A**
    B. Local variables for function **B**
    C. Address of function B
    D. Address inside **A** where **B** is called
    E. The return value for function **B**

13. (3 points) True or False? The physical memory allocated to a process at a given time must be contiguous.

    A. True **B. False**

14. (3 points) True or False? All memory accessible by a process is exclusively owned by that process.

    A. True **B. False**

Part III. Program Compilation

15. (6 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 = zformat
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:
14         $(CC) $(CFLAGS) part1.o part2.o -o zformat
15
16 clean:
17         rm *.o $(ALL)
```

A. 5 B. 8 **C. 13** D. 14 E. Answer not shown

16. (3 points) True or False? Assuming no bugs in the above Makefile, the *clean* command deletes all files that *make all* creates.

A. True B. False

17. (4 points) The *gcc* executable performs a number of different functions. Which one of these is responsible for combining multiple object files into a single, executable file?

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

Part IV. System Calls

18. (3 points) True or False? A process must be in kernel mode in order to allocate a new stack frame.

    A. True **B. False**

19. (5 points) A system call provides **detailed** error information back to the user program by doing what? (pick the best answer)

    A. Returning an error code
    B. Printing an error message to STDOUT
    C. Sending an error message to the system log
    D. Setting the global variable *errno*
    E. Answer not shown

20. (4 points) Which one of the following **is not** a sys call or sys call wrapper?

    A. fork() B. wait() C. open() D. read() **E. printf()**

21. (3 points) True or False? A process must be in kernel mode in order to output data to a USB stick.

    A. True B. False

Part V. Processes

22. (5 points) An active process that is not located in main memory is necessarily in what type of state?

A. Blocked B. Ready C. Running **D. Suspended** E. Answer not shown

23. (6 points) What is output by this program? Assume no errors in executing system calls.

```
int main(int argc, char** argv)
{
  for(int i = 1; i < 6; ++i) {
    int pid = fork();
    if(pid == 0) {
      i++;
      printf("%d", i);
      exit(0);
    }else{
      waitpid(pid);
    }
  }
}
```

A. 24 B. 246 C. 12345 **D. 23456** E. Answer not shown

24. (5 points) Which of the follow describes a critical section?

A. A block of memory that is reserved for the operating system
B. A block of memory that is reserved for a process
C. A block of memory that is shared between two processes
D. A sequence of operations that should not be interrupted
E. Answer not shown

25. (5 points) What information **is** kept in the Process Control Block? (select one)

A. Value of *errno*
B. Return address for the currently executing function
C. Return value for the currently executing function
D. Priority of the process
E. Global variable values

26. (5 points) In the following program, the parent process is to send an *int* to the child, which then prints out the value. On which line is there a bug?

```
1  int main(int argc, char** argv)
2  {
3    int filedes[2];
4
5    if(pipe(filedes) == -1) {
6      fprintf(stderr, "Error\n");
7      exit(-1);
8    }
9
10   int pid;
11   if((pid = fork()) == -1) {
12     fprintf(stderr, "Error\n");
13     exit(-1);
14   }else if(pid > 0){
15     close(filedes[1]);
16     int val = 42;
17     write(filedes[0], &val, sizeof(int));
18     sleep(1);
19   }else{
20     close(filedes[1]);
21     close(0);
22     dup2(filedes[0], 0);
23
24     int myval;
25     if(read(0, &myval, sizeof(int)) != sizeof(int)){
26       fprintf(stderr, "Error\n");
27       exit(-1);
28     }
29     printf("Got: %d\n", myval);
30   }
31 }
```

A. 15 B. 20 C. 22 D. 25 E. There is no bug

27. (3 points) True or False? As a result of a fork(), a child receives a complete copy of the parent's **process control block.**

A. True **B. False**

28. (5 points) What information **is not** kept in the Process Control Block? (select one)

A. Accounting information
B. ID number of the process
C. Address of memory storing the stack and heap
D. State of the processor registers
E. The current offset in an open file

29. (3 points) True or False? In the 5-state process model, a process in the **blocked state** can move directly to the **running state**.

A. True **B. False**

Part VI. Serial Processing

30. (6 points) What is output by this program when invoked as indicated?

```
int main(int argc, char **argv)
{
  char str[100];

  strncpy(str, "Hello, World", 13);
  strncpy(str, argv[1], 5);
  printf("%s\n", str);
}
```

```
./a.out Ciao
```

A. Ciao B. Ciaoo, World C. Hello, World
D. Hello, WorldCiao E. Answer not shown

31. (6 points) The following function is to interpret a sequence of zeros and ones as a binary number and return the corresponding value. It will continue to accept digits until a non-binary character is received.

For example: '1', '0', '1', ' ' will result in the value 5 being returned. And the sequence '1', '1', ' ' will result in the value 3 being returned.

```
1  long get_binary ()
2  {
3    char c;
4    int flag = 1;
5    long value = 0;
6
7    while(flag == 1)
8    {
9      c = getchar();
10     if(c >= '0' && c <= '1')
11     {
12       value = value * 10 + c - '0';
13     }
14     else
15     {
16       flag = 0;
17     }
18   }
19   return value;
20 }
```

On which line is the bug?

A. 10 **B. 12** C. 13 D. 17 E. There is no bug