CS 214 — Spring 2021 > ☑ Tests & Quizzes

Tests & Quizzes

Final Exam

Return to Assessment List

- For

Part 1 of 6 - C programming 2.0/ 8.0 Points

Question 1 of 31

Which statement about the length of int is incorrect? 1.0/ 1.0 Points

- ○ **A.** An int is longer than a char
- ✓ ○ **B.** An int contains at least 4 bytes
- ○ **C.** An int is at least as long as a short int
- ○ **D.** An int is no longer than a long int
- ○ **E.** All of the above are incorrect
- ○ **F.** None of the above are incorrect
- ○ **G.** I don't know

Answer Key: B (100.0% Correct), G (25.0% Correct)

Question 2 of 31

What should never appear in a header file? 0.0/ 1.0 Points

- ○ **A.** Macro definitions
- ○ **B.** Function prototypes
- ○ **C.** Function implementations
- ○ **D.** Type definitions
- ○ **E.** All of the above
- ✗ ○ **F.** None of the above

- ○ **G.** I don't know

Answer Key: C (100.0% Correct), G (25.0% Correct)

Question 3 of 31

Which keyword appears at the start of a macro definition? 1.0/ 1.0 Points

- ✓ ○ **A.** `#define`
- ○ **B.** `#macro`
- ○ **C.** `#begin`
- ○ **D.** `#defmacro`
- ○ **E.** All of the above
- ○ **F.** None of the above
- ○ **G.** I don't know

Answer Key: A (100.0% Correct), G (25.0% Correct)

Question 4 of 31

Which of the following is a way to create an array containing 5 ints? 0.0/ 1.0 Points

- ✗ ○ **A.** `int a[5] = malloc(5 * sizeof(int));`
- ○ **B.** `int a[] = malloc(5 * sizeof(int));`
- ○ **C.** `int *a = malloc(5 * int);`
- ○ **D.** `int *a = malloc(5 * sizeof(int));`
- ○ **E.** All of the above
- ○ **F.** None of the above
- ○ **G.** I don't know

Answer Key: D (100.0% Correct), G (25.0% Correct)

Question 5 of 31

Given: 0.0/ 1.0 Points

```
int x = 5;
int *p = &x;
```

Which of these code fragments does **not** have the same meaning as the others?

- ✖ ○ **A.** `*p = 6;`
- ○ **B.** `x = 6;`
- ○ **C.** `++p;`
- ○ **D.** `++x;`
- ○ **E.** All of the above are different
- ○ **F.** None of the above are different
- ○ **G.** I don't know

Answer Key: C (100.0% Correct), G (25.0% Correct)

Question 6 of 31

Given

0.0/ 1.0 Points

```
int *a = malloc(5 * sizeof(int));
```

On an architecture with 4-byte addresses and 4-byte ints, what is `sizeof(a)`?

- ○ **A.** 4
- ○ **B.** 5
- ✖ ○ **C.** 20
- ○ **D.** 160
- ○ **E.** All of the above
- ○ **F.** None of the above
- ○ **G.** I don't know

Answer Key: A (100.0% Correct), G (25.0% Correct)

Question 7 of 31

Given

0.0/ 1.0 Points

```
int a[5] = {1,2,3,4};
```

On an architecture with 4-byte addresses and 4-byte ints, what is `sizeof(a)`?

- ✖ ○ **A.** 4

- ○ **B.** 5
- ○ **C.** 16
- ○ **D.** 20
- ○ **E.** All of the above
- ○ **F.** None of the above
- ○ **G.** I don't know

Answer Key: D (100.0% Correct), G (25.0% Correct)

Question 8 of 31

Given

0.0/ 1.0 Points

```
int *p = malloc(20 * sizeof(int));
int *q = p + 10;
```

What will `free(q)` do?

- ○ **A.** Deallocate the array that p points to
- ✖ ○ **B.** Deallocate `p[10]`
- ○ **C.** Shrink p to contain 10 integers
- ○ **D.** Nothing
- ○ **E.** All of the above
- ○ **F.** None of the above
- ○ **G.** I don't know

Answer Key: F (100.0% Correct), G (25.0% Correct)

Part 2 of 6 - Files

1.25/ 4.0 Points

Question 9 of 31

Assume that the current working directory is `/User/xyz`. Assume that `/User/xyz` contains one subdirectory, `foo`. Choose the path that does **not** refer to this directory.

1.0/ 1.0 Points

- ○ **A.** `foo`
- ○ **B.** `/User/xyz/foo`

- ○ **C.** `../xyz/foo`
- ○ **D.** `./foo/../foo`
- ✓ ○ **E.** All refer to foo
- ○ **F.** None refer to foo
- ○ **G.** I don't know

Answer Key: E (100.0% Correct), G (25.0% Correct)

Question 10 of 31

Which of the following is **not** a property of the file system's inode structure? 0.0/ 1.0 Points

- ○ **A.** Size of inode is fixed
- ○ **B.** Minimal overhead for small files
- ✗ ○ **C.** Fast access to all data in a file
- ○ **D.** No maximum file size
- ○ **E.** All of the above are inode properties
- ○ **F.** None of the above are inode properties
- ○ **G.** I don't know

Answer Key: D (100.0% Correct), G (25.0% Correct)

Question 11 of 31

Assume that foo is a file in the working directory containing at least 100 bytes. Our program executes this code: 0.0/ 1.0 Points

```
int d, r;
char b[32];
d = open("foo", O_RDONLY);
r = read(d, b, 20);
```

Assuming that open() and read() succeed (`d != -1` and `r != -1`), what is the **minimum** number of bytes written to b?

- ○ **A.** 0
- ○ **B.** 1

- ✗ ○ **C.** 20
- ○ **D.** 32
- ○ **E.** None of the above
- ○ **F.** All of the above
- ○ **G.** I don't know

Answer Key: B (100.0% Correct), G (25.0% Correct)

Question 12 of 31

f is a file descriptor for an open file. We use `d = dup(f)` to create a second file descriptor for the same file. Assuming that dup() succeeds (`d != -1`), what must our program do to close this file? 0.25/ 1.0 Points

- ○ **A.** `close(f)`
- ○ **B.** `close(d)`
- ○ **C.** `close(f)` and `close(d)`
- ○ **D.** Either `close(f)` or `close(d)`, but not both
- ○ **E.** All of the above
- ○ **F.** None of the above
- ✓✗ ○ **G.** I don't know

Answer Key: C (100.0% Correct), G (25.0% Correct)

Part 3 of 6 - Unix command line 2.75/ 4.0 Points

Question 13 of 31

Choose the program that best matches each description 1.5/ 2.0 Points

A. cmp

B. cat

C. tee

D. diff

E. more

F. ps

G. grep

H. wc

✓ G 1. Print all lines matching a regular expression

✗ C 2. Check whether two files have the same content

✓ D 3. Print all differences between two text files

✓ H 4. Count the number of characters, words, and lines in a text file

✓ B 5. Print one or more files to standard output

✓ E 6. Print one or more files to standard output, one screenful at a time

✗ A 7. Read from standard input and write everything to standard output and a file

✓ F 8. Print information about running processes

Answer Key: 1:G, 2:A, 3:D, 4:H, 5:B, 6:E, 7:C, 8:F

Question 14 of 31

Assuming that foo and bar are existing programs, what will happen if we enter this shell command? 0.25/ 1.0 Points

```
foo | bar
```

- ○ **A.** The shell will execute foo, and then execute bar
- ○ **B.** The shell will execute foo, and then execute bar if foo reported success
- ○ **C.** The shell will execute foo and bar concurrently, sending the output of bar to the input of foo
- ○ **D.** The shell will execute foo and bar concurrently, sending the output of foo to the input of bar
- ○ **E.** All of the above
- ○ **F.** None of the above
- ✓✗ ○ **G.** I don't know

Answer Key: D (100.0% Correct), G (25.0% Correct)

Question 15 of 31

Assuming that foo is an existing program, and that bar is a file in the working directory, what will happen if we enter this shell command? 1.0/ 1.0 Points

```
foo > bar
```

- ○ **A.** The content of bar will be sent as the input to foo
- ✓ ○ **B.** The output of foo will replace the current contents of bar
- ○ **C.** The output of foo will be appended to the contents of bar

- ○ **D.** The shell will report an error
- ○ **E.** All of the above
- ○ **F.** None of the above
- ○ **G.** I don't know

Answer Key: B (100.0% Correct), G (25.0% Correct)

Part 4 of 6 - Processes 4.0/ 8.0 Points

Question 16 of 31

Process A uses fork() to create a child process, B. Under what circumstances will B become an orphan? 1.0/ 1.0 Points

- ✓ ○ **A.** A terminates without calling wait()
- ○ **B.** B terminates before A calls wait()
- ○ **C.** B terminates while A is blocked in a call to wait()
- ○ **D.** B is terminated early by a signal from A
- ○ **E.** All of the above
- ○ **F.** None of the above
- ○ **G.** I don't know

Answer Key: A (100.0% Correct), G (25.0% Correct)

Question 17 of 31

Process A uses fork() to create child process B and then calls wait(). B uses fork() to create child process C, and terminates. When C terminates, what will its status be? 0.0/ 1.0 Points

- ○ **A.** orphan process
- ✗ ○ **B.** zombie process
- ○ **C.** orphan zombie process
- ○ **D.** It depends on whether A calls wait() a second time
- ○ **E.** All of the above
- ○ **F.** None of the above
- ○ **G.** I don't know

Answer Key: C (100.0% Correct), G (25.0% Correct)

Question 18 of 31

We are writing a program that at some point will invoke another program and then resume when that program has completed. Which of these functions is **not** necessary to accomplish this? 0.0/ 1.0 Points

- ○ **A.** execl() or execv()
- ○ **B.** pipe()
- ○ **C.** fork()
- ○ **D.** wait()
- ✖ ○ **E.** All of the above are necessary
- ○ **F.** None of the above are necessary
- ○ **G.** I don't know

Answer Key: B (100.0% Correct), G (25.0% Correct)

Question 19 of 31

Which of the following signals can **not** be ignored? 1.0/ 1.0 Points

- ○ **A.** SIGTERM
- ✔ ○ **B.** SIGKILL
- ○ **C.** SIGINT
- ○ **D.** SIGHUP
- ○ **E.** All of the above can be ignored
- ○ **F.** None of the above can be ignored
- ○ **G.** I don't know

Answer Key: B (100.0% Correct), G (25.0% Correct)

Question 20 of 31

Process A has used fork() to create a child process, B. If process A receives SIGINT and terminates, what happens to process B? 0.0/ 1.0 Points

- ✖ ○ **A.** It halts with a segmentation fault

- ○ **B.** It receives SIGINT as well
- ○ **C.** It immediately terminates
- ○ **D.** It is unaffected
- ○ **E.** All of the above
- ○ **F.** None of the above
- ○ **G.** I don't know

Answer Key: D (100.0% Correct), G (25.0% Correct)

Question 21 of 31

Under what circumstances will execl() or execv() return? 0.0/ 1.0 Points

- ○ **A.** These functions return when the specified program has finished executing
- ✖ ○ **B.** These functions never return
- ○ **C.** These functions return if the specified program could not be executed
- ○ **D.** These functions return twice: once in the child, and once in the parent
- ○ **E.** All of the above
- ○ **F.** None of the above
- ○ **G.** I don't know

Answer Key: C (100.0% Correct), G (25.0% Correct)

Question 22 of 31

Process A opens a file, F, and then uses fork() to create a child process, B. If process B closes file F, what will happen? 1.0/ 1.0 Points

- ○ **A.** close() will fail because file F is not open in B
- ○ **B.** file F will be closed in both process A and process B
- ○ **C.** file F will be closed in process A, but remain open in process B
- ✔ ○ **D.** file F will be closed in process B, but remain open in process A
- ○ **E.** All of the above
- ○ **F.** None of the above
- ○ **G.** I don't know

Answer Key: D (100.0% Correct), G (25.0% Correct)

Question 23 of 31

Consider this code fragment (variable declarations omitted): 1.0/ 1.0 Points

```
d = 100;
c = fork();
if (c == 0) {
    d = 200;
    exit(EXIT_SUCCESS);
}
wait(NULL);
printf("%d\n", d);
```

Assuming that no errors occur, what will be printed?

- ✓ ○ **A.** 100
- ○ **B.** 200
- ○ **C.** Nothing
- ○ **D.** The behavior is not determined
- ○ **E.** All of the above
- ○ **F.** None of the above
- ○ **G.** I don't know

Answer Key: A (100.0% Correct), G (25.0% Correct)

Part 5 of 6 - Concurrency 2.25/ 5.0 Points

Question 24 of 31

Which of the following is **not** a precondition for deadlock? 1.0/ 1.0 Points

- ○ **A.** Mutual exclusion
- ○ **B.** Hold and wait
- ○ **C.** Circular wait
- ○ **D.** No pre-emption
- ✓ ○ **E.** All of the above are preconditions

- ○ **F.** None of the above are preconditions
- ○ **G.** I don't know

Answer Key: E (100.0% Correct), G (25.0% Correct)

Question 25 of 31

Which of the statements below is incorrect? 0.25/ 1.0 Points

- ○ **A.** A thread that waits twice for the same semaphore will deadlock
- ○ **B.** After waiting for a condition variable, a thread must re-check the desired condition
- ○ **C.** A mutex must only be unlocked by the thread that locked it
- ○ **D.** The number of threads that will wait at a barrier is specified when the barrier is created
- ○ **E.** All of the above are incorrect
- ○ **F.** None of the above are incorrect
- ✓✗ ○ **G.** I don't know

Answer Key: A (100.0% Correct), G (25.0% Correct)

Question 26 of 31

If our hardware architecture has no specialized concurrency-safe instructions, which synchronization mechanism can we implement? 0.0/ 1.0 Points

- ○ **A.** Semaphores
- ✗ ○ **B.** Mutexes
- ○ **C.** Condition variables
- ○ **D.** Barriers
- ○ **E.** All of the above
- ○ **F.** None of the above
- ○ **G.** I don't know

Answer Key: F (100.0% Correct), G (25.0% Correct)

Question 27 of 31

1.0/ 1.0 Points

We have organized our program into multiple threads, each of which performs some set up before it begins its primary work. Which synchronization method should we use to ensure no thread starts until every thread is ready?

- ○ **A.** Semaphore
- ✓ ○ **B.** Barrier
- ○ **C.** Mutex
- ○ **D.** Condition variable
- ○ **E.** All of the above
- ○ **F.** None of the above
- ○ **G.** I don't know

Answer Key: B (100.0% Correct), D (25.0% Correct), G (25.0% Correct)

Question 28 of 31

We are writing the pop() function for a thread-safe stack. If the stack is empty, we want pop() to block until the stack becomes non-empty. Assume that `head`, `nonempty`, and `lock` are pointers to the head of the stack, a condition variable, and a mutex, respectively. Assume that push() will call `pthread_signal(nonempty)` after adding something to the stack.

0.0/ 1.0 Points

Which of the following code fragments should we use after calling `pthread_mutex_lock(lock)`?

- ○ **G.** I don't know
- ○ **B.**

```
if (head == NULL) {
    pthread_cond_wait(nonempty, lock);
}
```

- ○ **E.** All of the above
- ○ **C.**

```
while (head == NULL) {
    pthread_cond_wait(nonempty, lock);
}
```

- ○ **F.** None of the above
- ○ **D.**

```
do {
    pthread_cond_wait(nonempty, lock);
} while (head == NULL);
```

- ✖ ○ **A.**

```
pthread_cond_wait(nonempty, lock);
```

Answer Key: G (25.0% Correct), C (100.0% Correct)

Part 6 of 6 - Network communication 2.43/ 4.0 Points

Question 29 of 31

If we are using the socket interface to write a server that accepts TCP connections, which of the functions below is **not** needed? 1.0/ 1.0 Points

- ○ **A.** socket()
- ○ **B.** accept()
- ○ **C.** listen()
- ○ **D.** bind()
- ✔ ○ **E.** All of the above are needed
- ○ **F.** None of the above are needed
- ○ **G.** I don't know

Answer Key: E (100.0% Correct), G (25.0% Correct)

Question 30 of 31

Which if these is **not** a feature of TCP connection sockets? 0.0/ 1.0 Points

- ○ **A.** Data lost in transmission will be resent
- ○ **B.** Data received out of order will be presented in order
- ○ **C.** Each call to read() will produce a complete message
- ○ **D.** read() can block if no data is available
- ✖ ○ **E.** All of the above are features of TCP connection sockets
- ○ **F.** None of the above are features of TCP connection sockets
- ○ **G.** I don't know

Answer Key: C (100.0% Correct), G (25.0% Correct)

Question 31 of 31

Match each OSI layer with its responsibility 1.43/ 2.0 Points

A. network

B. data link

C. session

D. application

E. physical

F. transport

G. presentation

✓ D 1. What sorts of messages are sent?

✓ G 2. How are messages encoded in bytes?

✓ C 3. Who establishes a connection? How many messages are sent per connection, and how many connections are established?

✓ F 4. How is a connection established? How is missing/out-of-order data handled?

✓ A 5. How is data passed between machines that may not be directly connected?

✗ E 6. How is data passed between machines that are directly connected?

✗ B 7. What hardware is used to transmit data?

Answer Key: 1:D, 2:G, 3:C, 4:F, 5:A, 6:B, 7:E

Student, Faculty and Staff: Contact the Office of Information Technology: help@oit.rutgers.edu: 833.OIT.HELP

Report an Accessibility Barrier

- Rutgers University
-