CS241 SP15 Exam 2: Solution Key

Name: Geng, Y. **UIN:** 656934543

Exam code: EABBED **NetID:** yangeng2

SCROLL TO THE NEXT PAGE TO REVIEW YOUR ANSWERS

A VERSION OF THESE QUESTIONS MAY APPEAR IN A FUTURE QUIZ

Name: Geng, Y netid: yangeng2 uin:656934543

1. (1 point.) Which one of the following is NOT true for a child process created by `fork`?

(A) Inherits (shares) the parent's open file streams

(B) Inherits signal handlers defined using `signal`

(C) Gets its own complete copy of the parent's process memory

(D) Starts after the parent process has finished

Correct answer: D.
Your answer: D.
1 out of **1** point received

2. (1 point.) A zombie is created when

(A) A parent sends the child process a `SIGZOMB` signal

(B) A parent doesn't wait on a finished child

(C) A child calls `exec`

(D) A child doesn't wait on a parent

(E) The parent finishes before the child process

Correct answer: B.
Your answer: B.
1 out of **1** point received

Solution. If the parent finishes first then the child's parent will become process 1 (init) which will correctly wait on processes.

3. (1 point.) Which one of the following best describes the code below?

```
int *p = (int *)malloc(sizeof(int));
p = NULL;
free(p);
```

(A) Memory leak

(B) The program may crash as `free()` is called on a NULL pointer

(C) There are no errors in the code

(D) Dangling pointer

(E) Compiler error: `free` cant be applied on NULL pointer

Correct answer: A.
Your answer: A.
1 out of **1** point received

Solution. The original allocation is never freed,so the code has a memory leak. Note, `free(NULL)` does nothing.

4. (1 point.) Which one of the following is NOT true for the following line?

```
waitpid(id,&status,0); // Assume waitpid is successful
```

(A) Is used to prevent child processes from starting immediately

(B) Is used to cleanup zombies from the kernel's process table

(C) If the child exited normally then `status` it includes the lowest 8 bits of the child's exit value

(D) The `status` variable can be queried with macros `WIFEXITED,WIFSIGNALED,WEXITSTATUS` to extract the status of the child process

(E) The status variable is an `int; &status` means the address of the integer

Correct answer: A.
Your answer: A.
1 out of **1** point received

5. (1 point.) Which one of the following is NOT true?

(A) The default action of `SIGALRM` is to quit the process

(B) Signals are software interrupts - they are handled concurrently by signal handlers.

(C) The default action of `SIGCHLD` is to do nothing

(D) Signals can be sent to other processes using the 'signal' function

(E) Pressing `CTRL-C` will send a `SIGINT` signal to the process

Correct answer: D.
Your answer: B.
0 out of **1** point received

Solution. Signals are sent to other processes using `kill`

6. (1 point.) Which one of the following is most likely to print `Hi` only once?

(A) `fork(); write(1,"Hi",2);`

(B) `execl("/bin/ls","ls",(char*)NULL); write(1,"Hi",2);`

(C) `execl("nosuchfile","nss",(char*)NULL); write(1,"Hi",2);`

(D) `puts("Hi");if(1) puts("Hi");`

Correct answer: C.
Your answer: B.
0 out of **1** point received

Solution. We need `exec` to fail! The incorrect response `execl("/bin/ls","ls",(char*)NULL);` will replace the running process with the `/bin/ls` program and the `write` call will never be executed.

7. (1 point.) What will be the most likely last thing printed by the following program?

```
1 int main() {
2    int c = fork();
3    printf("c=%d : pid=%d ppid=%d\n",c, getpid(),getppid() );
4    if(c>0) return 97;
5    sleep(4);
6    printf("Answer: %d\n",getppid());
7    return 80;
8  }
OUTPUT:
c=0 : pid=97 ppid=90
c=97 : pid=90 ppid=80
____ ?
```

(A) `Answer: 90`

(B) None of the other responses are correct

(C) `Answer: 80`

(D) `Answer: 1`

(E) `Answer: 97`

Correct answer: D.
Your answer: A.
0 out of **1** point received

Solution. When a process is orphaned because its parent has already finished it is adopted by `init` (process 1). `init` ensures there are no zombies (i.e. it will call `wait` or `waited` for every SIGCHLD signal).

8. (1 point.) How many times will `!` be printed when this program is run in a terminal?

```
int main() {
   printf("!");
   fork();
   fork();
   exit(0);
   return 0;
 }
```

(A) 2

(B) 3

(C) 5 or more

(D) 1

(E) 4

Correct answer: E.
Your answer: E.
1 out of **1** point received

Solution. The printf does not flush the buffer, so the process memory contains `!` in its buffer. The first `fork` creates two processes. The second `fork` is executed by both processes, so now there are four processes. At exit all four processes flush their buffer to the terminal.

9. (1 point.) If `malloc` fails (returns NULL) will the following program crash (seg fault)? If so, where?

```
1 void * ptr1 = (void*) malloc(16);
2 int ** ptr2 = (int**) ptr1;
3 int *** ptr3 = & ptr2;
4 void* ptr4= (void*) &ptr1;
```

(A) Line 3

(B) None of the other responses are correct

(C) Line 2

(D) Line 1

(E) Line 4

Correct answer: B.
Your answer: B.
1 out of **1** point received

Solution. `ptr1` is cast but never de-referenced: It is never used to attempt to read/write memory at address 0. The other lines of code get the address of the variable but do not read/write address held by of ptr1.

10. (1 point.) Which one of the following is the best description of POSIX process control? When a child process finishes (or temporarily stops) ...

(A) All siblings are notified with a `SIGQUIT` signal

(B) The child process is re-assigned a new parent process

(C) The process is automatically restarted

(D) The init (process 1) is sent a `SIGUSR1` signal

(E) The parent process is sent a `SIGCHLD` signal

Correct answer: E.
Your answer: E.
1 out of **1** point received

11. (1 point.) Which of the following is NOT true for `getline`?

(A) Is used to convert a character array into integer and floating point values

(B) It's important to set both capacity to zero and the character pointer to `NULL` before the first call to `getline`

(C) `getline` arguments include a pointer to an int and a pointer to a pointer to char, so it can modify their contents.

(D) `getline` returns the number of characters read (possibly including a newline character at the end)

(E) To avoid a memory leak, call free on the buffer after the last call to `getline`

Correct answer: A.
Your answer: A.
1 out of **1** point received

12. (1 point.) Which one of the following is true for `fork()` ?

(A) Creates a new process by cloning the existing process; the child starts by `fork()` returning 0

(B) Installs a new fork handler

(C) Creates a new process by reloading the program - the child starts at `main()`

(D) Creates a new file handle for standard int

(E) Creates a new file handle for standard out

Correct answer: A.
Your answer: A.
1 out of **1** point received

Solution. The cloning of the process happens at the moment of calling `fork`. The process is not restarted. Looking backwards in time, it is as if two processes showed the same history. Looking forwards in time, both processes have their own address space - changes to variables in one process will not affect the other.

13. (1 point.) Which response best describes the following buggy code that, when executed on a 32 bit machine (pointers require 4 bytes), is suppose to create a 16x16 2D character array?

```
1  char ** array;
2  array = (char**) malloc(16 * sizeof(char)); // bug here
3  if( ! array ) return 1; // malloc failed
4  int i = 0;
5  for(; i < 16;i++)
6    array[i] = malloc(16 * sizeof(16));
```

(A) Insufficient memory allocated (line 2) causes a buffer overflow when i is 12 and higher

(B) Insufficient memory allocated (line 2) causes a buffer overflow when i is 8 and higher

(C) Insufficient memory allocated (line 2) causes a buffer overflow when i is 0

(D) Insufficient memory allocated (line 2) causes a buffer overflow when i is 4 and higher

(E) Insufficient memory allocated (line 2) causes a buffer overflow when i is 16 and higher

Correct answer: D.
Your answer: D.
1 out of **1** point received

Solution. 16 bytes are allocated on the heap. On a 32 bit machine each pointer uses 4 bytes, thus there is sufficient space for 4 pointers. The fifth pointer (i=4) and higher, will be stored beyond the edge of the buffer.

14. (1 point.) When will `fork()` return -1 ?

(A) In the child process

(B) If `fork` failed

(C) When the parent is the first process

(D) When a child needs to be restarted

(E) In the parent process

Correct answer: B.
Your answer: B.
1 out of **1** point received

Solution. `fork` returns -1 (fail! No fork for you!) or: 0 in the child and a positive integer in the parent - so the parent can store the process id the newly created child.

15. (1 point.) `puts(ptr)` is equivalent to

(A) `fprintf(stderr,"%s\n",ptr)`

(B) `signal(SIGINT,ptr)`

(C) `ptr=getchar()`

(D) `printf("%s\n",ptr)`

(E) `scanf("%s\n",ptr)`

Correct answer: D.
Your answer: D.
1 out of **1** point received

16. (1 point.) Which one of the following is the best choice of missing code? Choose the correct snippet so that the program uses `string.h` functions to will display the message when the program is started with `"-h"` option

```
1  int main(int argc, char*argv[]) {
2    // If no arguments or just -h, then show a message and quit:
3    if(argc ==1 ||   _______    ) help_message_and_quit();
```

(A) `0==strcmp(argv[0],"-h")`

(B) `argv[1] == "-h"`

(C) `argv[0] == "-h"`

(D) `0==strcmp(argv[1],"-h")`

(E) `0==streq(argv[0],"-h")`

Correct answer: D.
Your answer: D.
1 out of **1** point received

Solution. `argv[0]` holds the program name. `strcmp` returns 0 if the two arguments are equal.

17. (1 point.) Four students were asked to write four alternative ways to print `Hello World!` to standard output. Carefully read the four functions below and for each one, decide if it will print `Hello World!` without error. Choose the most accurate response below.

```
void A() { char *s=(char*)malloc(100); strcpy(s,"Hello World!\n"); puts(s); free(s); }
void B() { char s[100]; *s=0; strcat(s, "Hello World!\n"); puts(s); }
void C() { static char s[100]; sprintf(s,"Hello "); strcat(s,"World!\n"); write(1,s,strlen(s)); }
void D() { char *s = "Hello "; strcat(s, "World!"); printf("%s\n", s); }
```

(A) 2 functions are correct

(B) Only 1 function is correct

(C) All 4 functions are correct

(D) None of the functions are correct

(E) 3 functions are correct

Correct answer: E.
Your answer: A.
0 out of **1** point received

Solution. `D()` will segfault at strcat because `s` points to readonly memory (the string constant).

18. (1 point.) Which one of the following changes the process's current directory to the user's home directory?

(A) None of the other responses are correct

(B) `pwd( environ[0] )`

(C) `pwd( environ[UHOME] )`

(D) `chdir( getenv("HOME") )`

(E) `chdir( environ[ getenv("HOME")] )`

Correct answer: D.
Your answer: E.
0 out of **1** point received

19. (1 point.) Which one of the following is true for `gets`?

(A) The function `gets` is the recommended function to read lines of input into a buffer

(B) Returns an integer pointer

(C) Allows a buffer overflow if the input line is longer than the buffer

(D) Can only read binary data from a file

(E) Can only read text data from a file

Correct answer: C.
Your answer: C.
1 out of **1** point received

20. (1 point.) Which one of the following is NOT true?

(A) `static` variables are automatic because they stored in the stack memory.

(B) The last entry of string arrays `argv` and `environ` is always `NULL`

(C) `malloc` allocates memory on the heap

(D) `argv[1]` is the first argument because `argv[0]` is the program name

(E) `char** environ` should be declared `extern`

Correct answer: A.
Your answer: A.
1 out of **1** point received

Solution. `static` variables are stored in the data segment, not the stack or heap. They are valid for the lifetime of the process.

21. (1 point.) Which one of the following is NOT true?

(A) The default SIGALRM handler does nothing

(B) If `open` succeeds it will return the smallest unused non-negative integer

(C) `alarm(5)` will send an asynchronous signal `SIGALRM` to the process in 5 seconds

(D) Dead processes (zombies) still take up space in the system's process table. If the system table is full no new processes can be created.

(E) Standard error stream (stderr, file descriptor 2) is not buffered

Correct answer: A.
Your answer: A.
1 out of **1** point received

Solution. The default action of SIGALRM is to kill the process.

22. (1 point.) Which response best describes the common system programming pattern to run another program and wait for it to finish?

(A) Parent process execs then child forks and child waits

(B) Child process waits then parent exec and child forks

(C) Parent process forks then child execs and parent waits

(D) Child process execs then child forks and child waits

(E) Parent process forks then parent execs and child waits

Correct answer: C.
Your answer: C.
1 out of **1** point received

Solution. Note, this common system programming pattern has the insightful name "fork-exec-wait."

23. (1 point.) Which one of the following prints `H` to the standard output stream?

```
1 char* ptr = "H";
2 ________?
```

(A) `write(sizeof(ptr), ptr, stdout);`

(B) `write(1,ptr,strlen(ptr));`

(C) `fprintf(stderr,"%s",ptr);`

(D) `printf("%p",ptr);`

(E) `puts(* ptr);`

Correct answer: B.
Your answer: B.
1 out of **1** point received

24. (1 point.) Which one of the following best describes the correct line 4 to read the line and store the result in the buffer and score variables?

```
1  char* buffer = (char*) malloc(16);
2  int score,res;
3  char* line = "Pointers 123";
4  _______  ?
```

(A) `res = sscanf(line, "%15s %d",buffer, &score);`

(B) `res = sscanf(line, "%15s %d",buffer, score);`

(C) `res = sscanf(line, "%15s %d", &buffer, &score);`

(D) `res = sscanf(line, "%15s %d", *buffer, *score);`

(E) `res = sscanf(line, "%15s %d", &buffer, score);`

Correct answer: A.
Your answer: A.
1 out of **1** point received

Solution. The `buffer` variable points to the memory that we wish `sscanf` to write into, whereas `score` is the integer that we wish to change, so we need its address.

```
> Did you bubble in your netid and UIN?
> Did you bubble in your exam key?
> Did you bubble in all questions?
> Did you write your name,netid and UIN on the exam?

> Please hand in your scantron with the response side 1-96 uppermost
> and Q1 in the top left corner
```

Summary of answers:

Question	Correct Answer	Your Answer	Points
1	D	D	1
2	B	B	1
3	A	A	1
4	A	A	1
5	D	B	0
6	C	B	0
7	D	A	0
8	E	E	1
9	B	B	1
10	E	E	1
11	A	A	1
12	A	A	1
13	D	D	1
14	B	B	1
15	D	D	1
16	D	D	1
17	E	A	0
18	D	E	0
19	C	C	1
20	A	A	1
21	A	A	1
22	C	C	1
23	B	B	1
24	A	A	1
Total			19