

## CS241 SP15 Exam 6: Solution Key

---

**Name:** Geng, Y.

**UIN:** 656934543

---

**Exam code:** BECAAC

**NetID:** yangeng2

---

SCROLL TO THE NEXT PAGE TO REVIEW YOUR ANSWERS

---

A VERSION OF THESE QUESTIONS MAY APPEAR IN A FUTURE QUIZ

---

1. (1 point.) Identify the missing the code at positions X,Y, and Z to create an unnamed pipe and write one byte into the pipe.

```
int fd[ _X_ ];
__Y__(fd);
// later...
write( fd[ _Z_ ] , "!",1);
```

- (A) X:2 Y:pipe Z:1
- (B) X:1 Y:open Z:0
- (C) None of the other responses are correct
- (D) X:2 Y:pipe Z:0
- (E) X:2 Y:mknod Z:1

**Correct answer:** A.

**Your answer:** A.

1 out of 1 point received

---

---

2. (1 point.) Which response best describes the following code segment?

```
int main() {
    FILE*fh=fopen("results.txt","w+");
    fprintf(fh, "%d",12345);
    fflush(fh);
    fseek( fh, 0, SEEK_SET);
    pid_t child = fork();
    if(child==0) { /* I'm the child */
        fseek( fh, 0, SEEK_END);
        fclose(fh);
        exit(0); // does not return
    }
    waitpid(child,NULL,0);

    fprintf(fh, "%d",0);
    fclose(fh);
    return 0;
}
```

- (A) The parent will never successfully write 0 to the file
- (B) 0 will be written at the end of the file
- (C) The child process will truncate the file to zero bytes
- (D) 0 will be written at the start of the file
- (E) The parent process will segfault because the file was already closed

**Correct answer:** B.

**Your answer:** B.

1 out of 1 point received

---

---

3. (1 point.) Which one of the following is the best description of POSIX process control? When a child process finishes (or temporarily stops) ...

- (A) The init (process 1) is sent a **SIGUSR1** signal
- (B) All siblings are notified with a **SIGQUIT** signal
- (C) The parent process is sent a **SIGCHLD** signal
- (D) The process is automatically restarted
- (E) The child process is re-assigned a new parent process

**Correct answer:** C.

**Your answer:** C.

1 out of 1 point received

---

---

4. (1 point.) Which one of the following is NOT an advantage of virtual memory?
- (A) To prevent fragmentation, sequential frames are assigned sequentially to pages
  - (B) Virtual memory allows processes to share read-only frames (e.g. C library, program code)
  - (C) Processes can share frames using the 'mmap' system call.
  - (D) There can be valid virtual addresses that do not have a physical memory assigned
  - (E) Stack memory can be set to be non-executable (i.e. only contain data)

**Correct answer:** A.

**Your answer:** A.

1 out of 1 point received

---

---

5. (1 point.) A 64 bit architecture with 16 GB of RAM uses 16 KB pages in a three-level page table. How many bits are used for the offset?

- (A) 20
- (B) 16
- (C) None of the other responses are correct
- (D) 8
- (E) 10

**Correct answer:** C.

**Your answer:** C.

1 out of 1 point received

---

**Solution.** 14

---

6. (1 point.) When will `fork()` return 0 ?

- (A) When the parent is the first process
- (B) If an error occurs
- (C) In the parent process
- (D) When a child needs to be restarted
- (E) In the child process

**Correct answer:** E.

**Your answer:** E.

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.

---

7. (1 point.) Which order of calls can be used to determine a file size (for files < 2GB)?

- (A) `fseek(fh,0,SEEK_END)` then `ftell(fh)`
- (B) `fset(fh)` then `fseek(fh,0,SEEK_SET)`
- (C) `fseek(fh,-1,SEEK_APP)` then `fpos(fh)`
- (D) `fpos(fh)` then `fseek(fh,-1,SEEK_APP)`
- (E) `fseekend(fh)` then `flength(fh)`

**Correct answer:** A.

**Your answer:** A.

1 out of 1 point received

---

---



8. (1 point.) Solve my riddle! I speed up the conversion of a virtual address to a physical address by caching recent results. I am useless if your memory requests are random (you'll need the page tables for that case) but usually your reads and writes are to recently used pages. My short-term memory is tiny but I am extremely fast! What am I called?

- (A) Translation Lookaside Buffer
- (B) Physical Address Cache
- (C) Memory Management Unit
- (D) Dynamic Ram Translation
- (E) Address Conversation Cache

**Correct answer:** A.

**Your answer:** A.

1 out of 1 point received

---

---

9. (1 point.) A pipe is an example of

- (A) MMU
- (B) APC
- (C) TLB
- (D) PAC
- (E) IPC

**Correct answer:** E.

**Your answer:** E.

1 out of 1 point received

---

---

10. (1 point.) Which one of the following is TRUE for a typical 32 bit hardware implementation of Virtual Memory? Assume the machine has 128MB of ram

- (A) A typical page size on a 32 bit linux machine is 32MB
- (B) The highest 12 bits of the virtual address are used as an offset
- (C) The page table converts frame numbers into offset numbers
- (D) The page table converts page numbers into offset numbers
- (E) A single-level page table is sufficient to fit into main memory

**Correct answer:** E.

**Your answer:** E.

1 out of 1 point received

---

---

11. (1 point.) Which one of the following might be used to re-read the first line of a file? Assume `fh` refers to a valid file handle and the line will be parsed using `fscanf` or `fgets`.

- (A) `frepo(fh,-1)`
- (B) `freadat(fh,0)`
- (C) `fseek(fh,0,SEEK_SET)`
- (D) `freread(fh)`
- (E) `fpos(fh)`

**Correct answer:** C.

**Your answer:** C.

1 out of 1 point received

---

---

12. (1 point.) A process performs many writes over its entire virtual memory space with no predictable pattern. On a machine that uses a single-level page table, the process would run \_\_\_ due to the additional overhead of virtual memory compared to an equivalent system with no virtual memory support.

- (A) 50% slower
- (B) None of the other responses are correct
- (C) 3x faster
- (D) 50% faster
- (E) 2x slower

**Correct answer:** E.

**Your answer:** E.

1 out of 1 point received

---

---

13. (1 point.) The page table includes a dirty bit for each frame. One purpose of this bit is ...
- (A) To determine if the frame is used by user processes or the kernel
  - (B) To determine if memory is being written by two processes
  - (C) To skip copying memory to secondary storage if the content is unchanged
  - (D) To avoid use of memory that has hardware errors detected during start-up
  - (E) To determine if the RAM frame corresponds to newly allocated heap memory

**Correct answer:** C.

**Your answer:** C.

1 out of 1 point received

---

---

14. (1 point.) How can you fix the following incorrect code so that the `append` function appends a comma and integer value to an open file and also restores the original file position before returning. You may assume the file remains  $< 2\text{GB}$

```
1 void append(FILE* f, int val) {  
2     fseek(f, 0, SEEK_END);  
3     long orig = ftell(f);  
4     fprintf(f,"%d",val);  
5     fseek(f, orig, SEEK_END);  
6 }
```

- (A) Line 5: Replace `SEEK_END` with `SEEK_CUR`
- (B) Line 4: Replace `fprintf` with `fwrite`
- (C) None of the other responses are correct
- (D) Swap lines 2 and 3. Line 5: `SEEK_END` should be `SEEK_SET`
- (E) Line 3: Replace `ftell` with `fposition`. Line 5: `SEEK_END` should be `SEEK_OFFSET`

**Correct answer:** C.

**Your answer:** D.

1 out of 1 point received

---

**Solution.** SP2015 regrade: Line 4 contained a typo (the above code is now fixed). There was no integer parameter to `fprintf`. Therefore “None of the other responses are correct” is also counted as a correct answer.

---

15. (1 point.) During a context switch, the current state of a process is saved so that execution can be resumed at a later time. Which one of the following is NOT true?

- (A) A context switch occurs when switching from the kernel code to a user process
- (B) All C library calls require a context switch
- (C) A hardware interrupt (e.g. timer interrupt) can cause a context switch
- (D) A context switch is required when a system call is made
- (E) A context switch occurs when a single-threaded process calls `read()` on an empty pipe

**Correct answer:** B.

**Your answer:** B.

1 out of 1 point received

---

---



16. (1 point.) Which one of the following is NOT TRUE for a hardware implementation of Virtual Memory?
- (A) Pages can be missing i.e. they may not have any corresponding physical memory associated with them
  - (B) The page table may store how recently a particular page was used
  - (C) The page table is stored in RAM
  - (D) The page table does not use the lowest bits of the virtual address
  - (E) The page table converts frame numbers into page numbers

**Correct answer:** E.

**Your answer:** E.

1 out of 1 point received

---

---

17. (1 point.) Which one of the following prints H to the standard output stream?

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

- (A) puts(\* ptr);
- (B) printf("%p",ptr);
- (C) write(1,ptr,strlen(ptr));
- (D) fprintf(stderr,"%s",ptr);
- (E) write(sizeof(ptr), ptr, stdout);

**Correct answer:** C.

**Your answer:** C.

1 out of 1 point received

---

---

18. (1 point.)

It is common to include the man section number with a call. For example, "fork(2)" "printf(3)" implies the discussion is about `fork` documented in the system-call section (section #2) of the man pages, while `printf` is documented in the C library (section #3) of the man pages. Choose the best response to, "Where would you expect to find `pipe` and why?"

- (A) `pipe(2)` because it works with two C library FILE objects
- (B) `pipe(3)` because it works with integer file descriptors
- (C) `pipe(3)` because it works with two C library FILE objects
- (D) `pipe(2)` because it works with integer file descriptors
- (E) None of the other responses are correct

**Correct answer:** D.

**Your answer:** D.

1 out of 1 point received

---

---

19. (1 point.) Spot the error! When run, the `f2` function causes a segfault during the `strcpy` call. Which response best describes the bug that caused the segfault? Assume the `calloc` call is successful. The declaration of `strcpy` is `char * strcpy(char *dest, const char *src);`.

```
1  pthread_t tid;
2
3  void* hello(void*m) {
4      strcpy(m, "Hello world");
5      return m;
6  }
7  void f2() {
8      void* mem=calloc(100, sizeof(char));
9      pthread_create(&tid,NULL,hello,mem);
10     free(mem);
11     pthread_join(tid,&result);
12 }
```

- (A) Line 10 and 11 need to be swapped
- (B) Line 11: `pthread_join` should be `pthread_exit`
- (C) `strcpy` can only be used in the main thread
- (D) Line 8 and 9 need to be swapped
- (E) The `calloc` call does not allocate sufficient memory

**Correct answer:** A.

**Your answer:** A.

1 out of 1 point received

---

**Solution.** Use-after-free error: The new thread uses the heap memory after it has been free'd

---

20. (1 point.) While working on the discussion section code, your friend describes their solution (in pseudo-code) to the dining philosophers problem: “To prevent deadlock, wait until you can take both chopsticks at the same time - see my pseudo-code below!” Assume `trylock` either locks an unlock mutex or immediately returns failed

eat:

```
while (hungry) { // I am the the i-th philosopher

    while( trylock(i)==failed )
        cond_wait

    lock( ( i + 1 ) % N ) // take both chopsticks at the same time

    while( hungry ) { // Eat unless my neighbors want my chopsticks
        eat_for_a_minute()
    }
    unlock( i )
    unlock( ( i + 1 ) % N )
    cond_broadcast
}
```

Which of the following best describes your friend’s solution?

- (A) Can deadlock if all philosophers are hungry at the same time
- (B) Will not deadlock because there is no mutual exclusion
- (C) Is a valid solution but only one philosopher can eat a time
- (D) Can suffer from starvation and livelock
- (E) Will not deadlock because hold-and-wait is not satisfied

**Correct answer:** A.

**Your answer:** A.

1 out of 1 point received

---

**Solution.** All philosophers can grab their left chopstick at the same time, resulting in deadlock

---

21. (1 point.) Which one of the following is NOT true for a multi-level page table?

- (A) For lookups into the same frame, the TLB will be faster at virtual address translation than a multi-level page table
- (B) Is faster than a single-level page table for virtual address translation
- (C) Useful for 64bit because it can be sparse; not all sub-tables need to exist
- (D) Like single-page tables, uses an offset for each frame to calculate the physical address
- (E) Can identify pages that have been modified compared to the copy on disk

**Correct answer:** B.

**Your answer:** B.

1 out of 1 point received

---

---

22. (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 4
- (D) Line 1
- (E) Line 2

**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`.

---

23. (1 point.) Spot the error(s)! 5 threads will call `barrier` once. The first 4 threads should block until the 5<sup>th</sup> thread calls `barrier`, then all 5 threads should continue. A student wrote the following code and wonders if it will work correctly. Carefully review the multi-threaded code below for synchronization errors. Note `PTHREAD_COND_INITIALIZER` is equivalent to `pthread_cond_init`.

```
01 int c=5;
02 pthread_mutex_t m = PTHREAD_MUTEX_INITIALIZER;
03 pthread_cond_t cv = PTHREAD_COND_INITIALIZER;
04
05 void barrier() {
06     pthread_mutex_lock(&m);
07     while(c > 0) {
08         pthread_cond_wait(&cv, &m);
09     }
10     c--;
11     pthread_cond_broadcast(&cv);
12     pthread_mutex_unlock(&m);
13 }
```

Decide if each statement is true or false and select the appropriate response.

S1: “The code suffers from a race condition if two or more threads call `barrier` at the same time.

S2: “It is possible that some threads can continue *before* the 5<sup>th</sup> thread calls `barrier`”

S3: “It is possible that all five threads get stuck inside the barrier function even *after* the 5<sup>th</sup> thread calls `barrier`.”

- (A) Only S2 is true
- (B) Only S1 is true
- (C) Only S3 is true
- (D) None of the other responses are correct
- (E) Exactly two statements are true

**Correct answer:** C.

**Your answer:** C.

1 out of 1 point received

---



24. (1 point.) A pipe will generate a POSIX signal (SIGPIPE) ...

- (A) When writing and the pipe is full but not when the pipe is empty
- (B) When all writers are closed and a read is attempted
- (C) When a reader or writer would block
- (D) When writing and all listeners (readers) are already closed
- (E) When reading and the pipe is empty but not when the pipe is full

**Correct answer:** D.

**Your answer:** D.

1 out of 1 point received

---

---

25. (1 point.) Which of the following is NOT true for `getline`?

- (A) `getline` arguments include a pointer to an int and a pointer to a pointer to char, so it can modify their contents.
- (B) It's important to set both capacity to zero and the character pointer to `NULL` before the first call to `getline`
- (C) Is used to convert a character array into integer and floating point values
- (D) To avoid a memory leak, call `free` on the buffer after the last call to `getline`
- (E) `getline` returns the number of characters read (possibly including a newline character at the end)

**Correct answer:** C.

**Your answer:** A.

0 out of 1 point received

---

---

26. (1 point.) In CS241, IPC stands for

- (A) Interrupted program counter
- (B) Interprocess communication
- (C) Inert pre-emptive Coffman
- (D) Interprocess cancelation
- (E) Infinite pre-emptive Condition

**Correct answer:** B.

**Your answer:** B.

1 out of 1 point received

---

- 
- > 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	A	A	1
2	B	B	1
3	C	C	1
4	A	A	1
5	C	C	1
6	E	E	1
7	A	A	1
8	A	A	1
9	E	E	1
10	E	E	1
11	C	C	1
12	E	E	1
13	C	C	1
14	C	D	1
15	B	B	1
16	E	E	1
17	C	C	1
18	D	D	1
19	A	A	1
20	A	A	1
21	B	B	1
22	B	B	1
23	C	C	1
24	D	D	1
25	C	A	0
26	B	B	1
<b>Total</b>			25