CS241 SP15 Exam 6: Solution Key

Name: Unni, N. **UIN:** 664350897

Exam code: ADBACB **NetID:** nunni2

SCROLL TO THE NEXT PAGE TO REVIEW YOUR ANSWERS

A VERSION OF THESE QUESTIONS MAY APPEAR IN A FUTURE QUIZ

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

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

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

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

2. (1 point.) A pipe will generate a POSIX signal (SIGPIPE) ...

(A) When a reader or writer would block

(B) When all writers are closed and a read is attempted

(C) When writing and the pipe is full but not when the pipe is empty

(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

3. (1 point.) Which one of the following is NOT true for a multi-level page table?

(A) Like single-page tables, uses an offset for each frame to calculate the physical address

(B) Can identify pages that have been modified compared to the copy on disk

(C) For lookups into the same frame, the TLB will be faster at virtual address translation than a multi-level page table

(D) Is faster than a single-level page table for virtual address translation

(E) Useful for 64bit because it can be sparse; not all sub-tables need to exist

4. (1 point.) The page table includes a dirty bit for each frame. One purpose of this bit is ...

(A) To determine if the RAM frame corresponds to newly allocated heap memory

(B) To avoid use of memory that has hardware errors detected during start-up

(C) To skip copying memory to secondary storage if the content is unchanged

(D) To determine if memory is being written by two processes

(E) To determine if the frame is used by user processes or the kernel

5. (1 point.) Which one of the following is NOT TRUE for a hardware implementation of Virtual Memory?

(A) The page table converts frame numbers into page numbers

(B) The page table does not use the lowest bits of the virtual address

(C) Pages can be missing i.e. they may not have any corresponding physical memory associated with them

(D) The page table is stored in RAM

(E) The page table may store how recently a particular page was used

6. (1 point.) A process performs many writes over it's 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) 3x faster

(C) 2x slower

(D) 50% faster

(E) None of the other responses are correct

7. (1 point.) When will `fork()` return -1 ?

(A) When the parent is the first process

(B) When a child needs to be restarted

(C) In the parent process

(D) In the child process

(E) If `fork` failed

8. (1 point.) In CS241, IPC stands for

(A) Infinite pre-emptive Condition

(B) Interprocess communication

(C) Inert pre-emptive Coffman

(D) Interrupted program counter

(E) Interprocess cancelation

9. (1 point.) A 64 bit architecture with 1 GB of RAM uses 1 KB pages in a three-level page table. How many bits are used for the offset?

(A) None of the other responses are correct

(B) 14

(C) 20

(D) 10

(E) 16

10. (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) All C library calls require a context switch

(B) A context switch occurs when a single-threaded process calls `read()` on an empty pipe

(C) A context switch is required when a system call is made

(D) A hardware interrupt (e.g. timer interrupt) can cause a context switch

(E) A context switch occurs when switching from the kernel code to a user process

11. (1 point.) A pipe is an example of

(A) APC

(B) PAC

(C) TLB

(D) IPC

(E) MMU

12. (1 point.) Which one of the following is NOT an advantage of virtual memory?

(A) Stack memory can be set to be non-executable (i.e. only contain data)

(B) Virtual memory allows processes to share read-only frames (e.g. C library, program code)

(C) There can be valid virtual addresses that do not have a physical memory assigned

(D) To prevent fragmentation, sequential frames are assigned sequentially to pages

(E) Processes can share frames using the 'mmap' system call.

13. (1 point.) Spot the error(s)! 5 threads will call `barrier` once. The first 4 threads should block until the 5^{th} 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 count=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    count--;
08    pthread_cond_broadcast(&cv);
09    while(count > 0)
10              pthread_cond_wait(&cv, &m);
11    pthread_mutex_unlock(&m);
12  }
```

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^{th} thread calls `barrier`"
$S3$: "It is possible that one or more of the first four threads may get stuck inside the barrier function even *after* the 5^{th} thread calls `barrier`."

(A) Exactly two statements are true

(B) Only $S1$ is true

(C) Only $S2$ is true

(D) Only $S3$ is true

(E) None of the other responses are correct

14. (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) The highest 12 bits of the virtual address are used as an offset

(B) A typical page size on a 32 bit linux machine is 32MB

(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

15. (1 point.) Which one of the following prints `H` to the standard output stream?

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

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

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

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

(D) `puts(* ptr);`

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

16. (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) `freadat(fh,0)`

(B) `freread(fh)`

(C) `fpos(fh)`

(D) `fseek(fh,0,SEEK_SET)`

(E) `frepo(fh,-1)`

17. (1 point.) Which response best describes the following code segment?

```
int main() {
  FILE* fh = fopen("data.txt","w+");
  fprintf(fh, "--ABCD--");
  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, "@");
  fclose(fh);
  return 0;
}
```

(A) The parent will never successfully write `@` to the file

(B) `@` will be written at the start of the file

(C) `@` will be written at the end of the file

(D) The child process will truncate the file to zero bytes

(E) The parent process will segfault because the file was already closed

18. (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) None of the other responses are correct

(B) `Answer: 1`

(C) `Answer: 90`

(D) `Answer: 97`

(E) `Answer: 80`

19. (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:2 Y:pipe Z:0`

(C) `X:2 Y:mkfifo Z:1`

(D) `X:1 Y:open Z:0`

(E) None of the other responses are correct

20. (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 < 2GB

```
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) Line 3: Replace `ftell` with `fposition`. Line 5: `SEEK_END` should be `SEEK_OFFSET`

(D) None of the other responses are correct

(E) Swap lines 2 and 3. Line 5: `SEEK_END` should be `SEEK_SET`

21. (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) Address Conversation Cache

(B) Translation Lookaside Buffer

(C) Physical Address Cache

(D) Memory Management Unit

(E) Dynamic Ram Translation

22. (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 two C library FILE objects

(C) `pipe(3)` because it works with integer file descriptors

(D) None of the other responses are correct

(E) `pipe(2)` because it works with integer file descriptors

23. (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 init (process 1) is sent a `SIGUSR1` 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

24. (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) `fpos(fh)` then `fseek(fh,-1,SEEK_APP)`

(C) `fseekend(fh)` then `flength(fh)`

(D) `fseek(fh,-1,SEEK_APP)` then `fpos(fh)`

(E) `fset(fh)` then `fseek(fh,0,SEEK_SET)`

25. (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) Will not deadlock because there is no mutual exclusion

(B) Can deadlock if all philosophers are hungry at the same time

(C) Can suffer from starvation and livelock

(D) Is a valid solution but only one philosopher can eat a time

(E) Will not deadlock because hold-and-wait is not satisfied

26. (1 point.) Spot the error! When run, the `go` function causes a segfault during the `qsort` call. Assume `comp_fn` is correctly written and the `calloc` call is successful. Which response best describes the bug that caused the segfault?

```
1    pthread_t tid;
2    void* result;
3    void* func(void*m) {
4       qsort(m, 100000, sizeof(int), comp_fn);
5       return NULL;
6    }
7    void go() {
8       void* mem=calloc(100000, sizeof(int));
9       pthread_create(&tid,NULL,func,mem);
10      free(mem);
11      pthread_join(tid,&result);
12   }
```

(A) Line 8 and 9 need to be swapped

(B) Line 10 and 11 need to be swapped

(C) Line 11: `pthread_join` should be `pthread_exit`

(D) `qsort` must not be called in a second thread

(E) `qsort` can not be used with heap memory

Summary of answers:

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