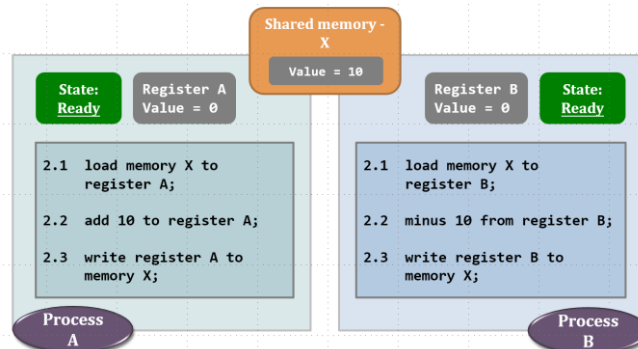


P1 Multiple Choices (30 points, 2 for each question)

1. Which is (are) not the function of operating system?
A. CPU management B. Storage management
☒ C. Network management D. Data management
2. What is(are) not shared among different threads in a process?
A. PID B. File descriptor
☒ C. Execution Stack ☒ D. Program Counter
3. Which system calls are employed to implement a C library call "system()"?
A. fork() B. exec*()
C. wait() D. pipe()
4. Pipe is one of process communication methods, which statement(s) is(are) NOT correct?
A. Pipe can transfer data in both directions
B. The pipe buffer size is decided by disk size
C. Either process read or write operation could be blocked during pipe-based communication
D. Pipe cannot be accessed by read and write processes at the same time.
5. Consider the lifecycle of a process, which status transferring is(are) impossible?
A. Ready → Running B. Running → Ready
C. Running → Waiting D. Ready → Waiting
6. Consider different job scheduling algorithms in OS, we should use () if shortest job has highest priority, () if the emergency job has highest priority, () if users want to improve the interactivity of the system, and linux scheduler employs ().
I. First Come First Service scheduling
II. Shortest Job First scheduling
III. Round-Robin scheduling
IV. Multiple Queue Priority scheduling
V. Preemptive Priority scheduling
A. II, V, I, IV B. I, V, III, IV C. I, II, IV, III D. II, V, III, IV.
7. Suppose there are three jobs J1, J2, J3 arrive at time step 0. The execution cost are T1, T2, and T3 (suppose $T1 < T2 < T3$). The scheduling algorithm is shortest job first, what is the average turnaround time ()
A. $T1+T2+T3$ B. $(T1+T2+T3)/3$ C. $(3T1+2T2+T3)/3$ D. $(T1+2T2+3T3)/3$
8. Given a critical section, we use "mutex" to guarantee mutual exclusion, when mutex=0, it means ().
A. There are two processes in critical section
B. Both processes are not in critical section

- C. One process is in critical section, the other is not in critical section
 D. One process is in critical section

9. What are the possible x value after the two processes executed () ?



- A. -10 and 30, B. 10 C. 0, 10, 20, D. -10, 0, 10, 20
10. Suppose we have 4 pages, the sequence of visited pages are 2, 0, 2, 9, 3, 4, 2, 8, 2, 4, 8, 4, 5. If the next visiting page is 7, which page will be evicted if we use LRU algorithm ().
 A. 2. B. 3. C. 4. D. 8
11. Suppose the available memory size is 55MB, we use Best Fit algorithm to allocate memory for each process. The allocation and release sequence are: (1) allocate 15MB, (2) allocate 30MB. (3) release 15MB. (4) allocate 8MB. (5) allocate 6MB. The size of the largest available memory is ().
 A. 7MB B. 9MB C. 10MB D. 15MB
12. () is (are) disk scheduling algorithm(s).
 A. Round robin B. LRU C. Shortest Seek Time First D. High priority first
13. Given HDD X, its average rotation cost is r seconds per turn, average seek time is T seconds. Each track could store N bytes. All sectors in the same track is a cluster (i.e., cluster is transfer unit). What is the cost to read b bytes () from HDD X?
 A. $(r+T)b/N$ B. b/NT C. $(b/N+T)r$ D. $bT/N+r$
14. For File System X, the cluster (block) size is 1KB, and the sector size is 512B. What is the disk size for a file with 1026B ()?
 A. 1026B B. 1536B. C. 1538B D. 2048B
15. User program issues disk I/O request, the I/O request will be processed as (1) user program → kernel I/O subsystem → device driver → Interrupt handler. Which phase calculates the cylinder, track and sector of the read/write data?
 A. user program B. kernel I/O subsystem C. device driver D. interrupt handler

P2 [Operating System] 16 points

(a) [3 pts] List the job scheduling algorithms you have learned in this course for process management

SJF / Round Robin /

(b) [3 pts] List the page replacement algorithms you have learned in this course about memory management

(c) [3 pts] What are the cons of existing file systems FAT32? Briefly explain the reasons:

空间不足, 随机访问比较差

(d) [3 pts] Compare the memory organization schemes of contiguous memory allocation, pure segmentation, and pure paging with respect to the following issues: (a) external fragmentation, (2) internal fragmentation, (3) ability to share code across processes.

dual mode

(e) [4 pts] In CS302, we have studied (1) Process management, (2) Memory management, (3) I/O device and (4) File system. Please write down three important concepts in each category from your perspective.

(1) process / thread /

(2) virtual address /

(3)

(4) file / NTFS /

P3 (16 points) System call

(a) [4pts] Assume that we have the following piece of code:

```
1. int main() {
2.   printf("Starting main\n");
3.   int file_fd = open("test.txt", O_WRONLY | O_CREAT, 0666);
4.   dup2(file_fd, STDOUT_FILENO);
5.   pid_t child_pid = fork();
6.   if (child_pid != 0) {
7.       wait(NULL);
8.       printf("CS302 SUSTech, in parent\n");
9.   } else {
10.      printf("CS302 SUSTech, in child\n");
11.  }
12. printf("Ending main: %d\n", child_pid);
13.}
```

What is the output of this program? You can assume that no syscalls failed and that *the child's PID is 123*. Fill in the following table with your prediction of the output:

Standard out	test.txt
Starting main	CS302 SUSTech, in parent ending main: 0

(b) [4pts] Next, assume that we have altered this code as follows:

```
1. int main() {
2.   printf("Starting main\n");
3.   int file_fd = open("test.txt", O_WRONLY | O_CREAT, 0666);
4.   int new_fd = dup(STDOUT_FILENO);
5.   dup2(file_fd, STDOUT_FILENO);
6.   pid_t child_pid = fork();
7.   if (child_pid != 0) {
8.       wait(NULL);
9.       printf("CS302 SUSTech, in parent \n");
10.  } else {
11.      dup2(new_fd, STDOUT_FILENO);
12.      printf("CS302 SUSTech, in child \n");
13.  }
14.  printf("Ending main: %d\n", child_pid);
15.}
```

What is the output this time? Fill in the following table with your prediction of the output:

Standard out	test.txt

(c) [4pts] Consider the following fragment of a program (which is missing lines of code). Fill in the blanks in the code to ensure that the output of this program is always the following two lines in the following order (under all interleavings or schedules):

```
val = 0
```

```
val = 1
```

No more than one function call (or system call) per blank! Your solution is not allowed to make assignments to the “val” variable. Also, no use of printf() or other print statements!

```
void apple () {
    pid_t oval = _____;
    kill(oval, SIGTEST);
}
void orange() {
    val = 1;
}
int val = 0;
int main() {
    pid_t pid = fork();
    _____;

    if (pid == 0) {
        _____;

    } else {
        _____;
    }
    printf("val=%d\n", val);
}
```

(d) [2pts] If a child process sends a SIGKILL signal to its parent, what happens to the parent process and the child process? Can the parent prevent this from happening?

(e) [2pts] Consider the following scenario. A process forks a child process and the parent process waits on it. Then, the child exits normally and the parent is unblocked. Finally, the parent makes another wait call on the child process. What happens to this last wait call and to the parent process?

P4 [Process Management] 16 points

(a) [8 pts] Consider the following snapshot of a system

	Allocation				Max				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P0	0	0	1	2	0	0	1	2	1	5	2	0
P1	1	0	0	0	1	7	5	0				
P2	1	3	5	4	2	3	5	6				
P3	0	0	1	4	0	6	5	6				

Answer the following questions using the banker's algorithm:

- What is the content of the matrix **Need**?
- Is the system in a safe state? If yes, please give a safe sequence. *Yes, (0,2,1,3)*
- If a request from process P1 arrives for (0,4,2,0), can the request be granted immediately?
Please explain your answer. *Yes (0,2,1,3)*

(b) [8 pts] The sleeping teaching assistant at CS302 (answer it with pseudocode)

CS302 has a teaching assistant (TA) who helps CS302 students with their project assignment during regular office hours. The TA's office is rather small and has room for only one desk with a chair and computer. There are n chairs in the hallway outside the office where students can sit and wait if the TA is currently helping another student. When there are no students who need help during office hours the TA sits at the desk and take a nap. If a student arrives during office hours and finds TA sleeping, the student must awaken the TA to ask for help. If a student arrives and finds TA currently helping another student, the student sits on one of the chairs in the hallway and waits. If no chairs are available, the student will come back later.

Using mutex locks and semaphores implement a solution that coordinates the activities of the TA and the students. (Full points will only give the optimal solution)

P5 [Memory Management] 16 points

(a) Demand paging. Given the page table of Process X as follows.

Virtual Page No.	Physical Page Frame	Valid ?
0	101H	1
1	302H	0
2	254H	1

Page size is 4KB, the memory access latency is 100ns, the TLB access latency is 10ns, the page fault processing cost is 10^8 ns (it includes the cost to update TLB and page table). The number of page frames is 2. The page replacement policy is LRU (least recently used algorithm). Assuming that

- 1) TLB is empty at the beginning
- 2) For address translation, MMU visits TLB first, if TLB miss then visits page table.
- 3) Valid = 0 indicates page is not in memory, incurs page faults. After handled page faults, the process will re-execute this instruction.

Given the virtual memory access sequence 2362H, 1565H, 25A5H.

- (1) [6 pts] What is the access latency for each above memory?
- (2) [2 pts] What is the corresponding physical address of virtual memory address 1565H, and why?

(b) Consider a system with following specifications:

Both virtual address space and physical address are 32 bits

Page table entry size of 4 Bytes

- (1) [4 pts] Suppose it uses 1-level page table, the format of the translation scheme is

<i>20 bit page</i>	<i>12 bit offset</i>
--------------------	----------------------

What is the page size? What is the maximum page table size?

- (2) [4 pts] Suppose it uses 2-level page table, the format of the translation scheme is

<i>10 bit page</i>	<i>10 bit page</i>	<i>12 bit offset</i>
--------------------	--------------------	----------------------

Please write down the 1-st level page number and its offset in decimal (base 10) of virtual address C302C302 (base 16).

Please write down the 2-nd level page number and its offset in decimal (base 10) of virtual address EC6666AB (base 16).

P6 [File System and I/O] 16 points

- (a) [2 pts] The FAT file system eliminates external fragmentation by allocating disk space in block-sized units. Considering that fact, briefly explain why defragmentation is still sometimes necessary.
- (b) [2 pts] In the UNIX 4.2 BSD FFS, inodes have direct pointers, a singly indirect pointer, a doubly indirect pointer, and a triply indirect pointer. The maximum file size supported by this inode type is approximately the same as the maximum file size supported by an inode with only a triply indirect pointer. Briefly explain one disadvantage of an inode design that only uses a triply indirect pointer instead of a combination of pointers (*other than the slightly reduced maximum file size*).
- (c) [10 pts] File F has 200 blocks (start from block 1). User A wants to insert one block to File F as its block 30. Please answer the following questions and briefly explain your answer.
- (1) [3pts] Suppose the file system employs Contiguous allocation. As shown in following figure, the gray blocks are free. How many disk block accesses you need to insert it at least, what should be updated in root directory?



- (2) [3pts + 4 pts] Suppose the file system employ linked allocation, each block has a link pointer and data (as shown in following figures). How many disk block accesses you need to insert it? If the size of each block is 1KB and the pointer size is 4 bytes. What is the maximum file size in this file system?



- (d) [2 pts] Please describe the file deletion procedure in FAT file system.