Sheet1 - Solution
Process Description and Control

1) What is an instruction trace?

   An instruction trace for a program is the sequence of instructions that execute for that process.

2) What common events lead to the creation of a process?
   - New batch job;
   - interactive logon;
   - created by OS to provide a service;
   - spawned by existing process.

3) For the 5-state process model, briefly define each state.
   - Running: The process that is currently being executed.
   - Ready: A process that is prepared to execute when given the opportunity.
   - Blocked: A process that cannot execute until some event occurs, such as the completion of an I/O operation.
   - New: A process that has just been created but has not yet been admitted to the pool of executable processes by the operating system.
   - Exit: A process that has been released from the pool of executable processes by the operating system, either because it halted or because it aborted for some reason.

4) What is swapping and what is its purpose?

   Swapping involves moving part or all of a process from main memory to disk. When none of the processes in main memory is in the Ready state, the operating system swaps one of the blocked processes out onto disk into a suspend queue, so that another process may be brought into main memory to execute.

5) In the 7-state process model, why do we have two blocked states?

   There are two independent concepts: whether a process is waiting on an event (blocked or not), and whether a process has been swapped out of main memory (suspended or not). To accommodate this 2 *2 combination, we need two Ready states and two Blocked states.

6) List four characteristics of a suspended process.
   - The process is not immediately available for execution.
   - The process may or may not be waiting on an event. If it is, this blocked condition is independent of the suspend condition, and occurrence of the blocking event does not enable the process to be executed.
   - The process was placed in a suspended state by an agent; either itself, a parent process, or the operating system, for the purpose of preventing its execution.
   - The process may not be removed from this state until the agent explicitly orders the removal.

7) For what types of entities does the OS maintain tables of information for management purposes?

The OS maintains tables for entities related to memory, I/O, files, and processes.

8) What are the elements of a process image?

Each process image consists of:

- A process control block,
- A user stack,
- The private address space of the process (program and data), and
- Any other address space that the process shares with other processes.

9) List three general categories of information in a process control block.

- Process identification
- Processor state information
- Process control information.

10) Why are two modes (user and kernel) needed?

The user mode has restrictions on the instructions that can be executed and the memory areas that can be accessed. This is to protect the operating system from damage or alteration. In kernel mode, the operating system does not have these restrictions, so that it can perform its tasks.

11) What are the steps performed by an OS to create a new process?

- Assign a unique process identifier to the new process.
- Allocate space for the process.
- Initialize the process control block.
- Set the appropriate linkages.
- Create or expand other data structures.

12) What is the difference between an interrupt and a trap?

An interrupt is due to some sort of event that is external to and independent of the currently running process, such as the completion of an I/O operation. A trap relates to an error or exception condition generated within the currently running process, such as an illegal file access attempt.

13) Give three examples of an interrupt.

Clock interrupt, I/O interrupt, memory fault.

14) What is the difference between a mode switch and a process switch?

A mode switch may occur without changing the state of the process that is currently in the Running state. A process switch involves taking the currently executing process out of the Running state in favor of another process. The process switch involves saving more state information.

15) You have executed the following C program:

```
main() {
        int pid;
        pid = fork();
        printf ("%d \n", pid);
}
```

What are the possible outputs?

On success, the PID of the child process is returned by `fork()` in the parent, and 0 is returned in the child. On failure, -1 is returned in the parent, no child process is created, and errno is set appropriately.

16) The ability of one process to spawn a new process is an important capability, but it is not without its dangers. Consider the consequences of allowing a user to run the process in the following figure. Assume that `fork()` is a system call that spawns a child process.

```
main() {
        while(true) {
                fork();
        }
}
```

a) Assuming that a system allowed such a process to run, what would the consequences be?

Exponential growth of processes occurs, while system has only a finite number of process ID's and a finite amount of memory. However since the size of created processes is small and since the OS has lots of swap space available the given code will most likely exhaust process table (run out of IDs) instead of run out of memory. The system will crash and the only choice is to re-boot the system.

b) Suppose you decide that it is inappropriate to reject certain processes, and that the best approach is to place certain runtime controls on them. What controls might the operating system use to detect processes like the above at runtime?

OS can place a pre-specified limit on the number of processes a user can create. Also, it may limit the number of child processes spawned by a certain process. If this limit is exceeded, the process should be terminated or the system administrator is notified.