INSTRUCTIONS: *If any question is unclear or you believe some assumptions need to be made, state your assumptions clearly at the beginning of your answer.*

---

1. (10%) What would be the output of the following C program? (*Note that the line numbers are for reference only.*)

```
1  #include <stdio.h>
2  #include <unistd.h>
3  #include <sys/types.h>
4  #include <sys/wait.h>
5  int main()
6  {
7      int status, fd[2];
8      pipe(fd);
9      pid_t pid = fork();
10     if (pid > 0) {
11         close(fd[1]);
12         close(0);
13         dup(fd[0]);
14         close(fd[0]);
15         waitpid(-1, &status, 0);
16         char buf[128];
17         int n = scanf("%s", buf);
18         printf("%d:%s\n", n, buf);
19     }
20     else if (pid == 0) {
21         close(fd[0]);
22         close(1);
23         dup(fd[1]);
24         close(fd[1]);
25         execl("/bin/echo", "echo", "welcome", "to", "nsysu", (void*) 0);
26     }
27     else {
28         return 1;
29     }
30     return 0;
31 }
```

2. (10%; 5% each) Assume a page reference string for a process with m frames (initially all empty). The page reference string has length n with p distinct page numbers occurring in it. For any page-replacement algorithms,

    (a) What is an upper bound on the number of page faults? *Justify your answer for credit.*

    (b) What is a lower bound on the number of page faults? *Justify your answer for credit.*

3. (10%) A computer whose processes have 4096 pages in their address spaces keeps its page tables in memory. The overhead required for reading a word from the page table is 600 nsec. To reduce this overhead, the computer has a TLB, which holds 64 (page, frame) pairs and can do a lookup in 100 nsec. What hit rate is needed to reduce the mean overhead to 200 nsec or less, *assuming that TLB is looked up first before the page table is read*?

4. (10%; 5% each) A disk has 24000 cylinders, each with 16 tracks of 1024 blocks. A seek takes 1 ms per cylinder moved. If no attempt is made to put the blocks of a file close to each other, two blocks that are logically consecutive (i.e., follow one another in the file) will require an average seek, which takes 6 ms. If, however, the operating system makes an attempt to cluster related blocks, the mean interblock distance can be reduced to 2 cylinders and the seek time reduced to 200 μs. Assuming that the rotational latency is 3 ms and the transfer time is 10 μs per block, answer the following questions.

   (a) How long does it take to read a 400 block randomly placed file?

   (b) How long does it take to read a 400 block clustered file?

5. (10%) A small computer has four page frames. At the first clock tick, the R bits are 0111 (page 0 is 0, the rest are 1). At subsequent clock ticks, the values are 1011, 1010, 1101, 0010, 1010, 1100, 0001, 0101, 1011, and 1001. If the aging algorithm is used with an 8-bit counter, give the values of the four counters after the last ticks.

6. (20%; 4% each) Consider a file currently consisting of 150 blocks. Assume that the file control block is already in memory. Calculate how many disk I/O operations are required for contiguous and linked allocation strategies, if, for one block, the following conditions hold. In the contiguous allocation case, assume that there is no room to grow in the beginning, but there is room to grow in the end. Assume that the block information to be added is stored in memory.

   (a) The block is removed from the beginning.

   (b) The block is removed from the end.

   (c) The block is added at the beginning.

   (d) The block is added in the middle.

   (e) The block is added at the end.

7. (10%) Consider the interprocess-communication scheme where mailboxes are used. Suppose a process P wants to wait for two messages, one from mailbox A and one from mailbox B. What sequence of send and receive should it execute so that the messages can be received in any order without being blocked?

8. (10%; 5% each) Suppose that a scheduler has k ready processes at time 0 and that no new processes are created after time 0. Process i ($0 < i \leq k$) requires i units of computing time. Answer each of the following questions.

   (a) For a preemptive, round-robin scheduler with a scheduling quantum of one time unit, what is the mean turnaround time for these processes, assuming that process k is at the front of the ready queue and that other processes appear in decreasing order of required computing time?

   (b) For a non-preemptive, shortest-job-first scheduler, what is the mean turnaround time for these processes?

9. (10%) Suppose that a 32-bit virtual address is broken up into four fields, a, b, c, and d. The first three are used for a three-level page table system. The fourth field, d, is the offset. Does the number of pages depend on the sizes of all four fields? If not, which ones matter and which ones do not?