Problem 1 (0 Points)
Tanenbaum, Problem 40 (Chapter 2).

Problem 2 (2 Points)

a) Consider the case of two tasks \( (N = 2) \) where the service demands are \( t(1) = 5 \) and \( t(2) = 3 \). What is the average response time for the two possible service orderings (i.e., job 1 then job 2 and job 2 then job 1)?

b) Consider the more general case of \( N \) jobs with service demands of \( t(i) \), \( i=1:N \) where the service demands are strictly ordered; i.e., \( t(1) < t(2) < \ldots < t(N) \). Prove that SJF will result in the smallest average response time. HINT: Consider the average response time when Job i swaps service position with Job i-1.

Problem 3 (6 Points)

We wish to derive the equations of motion for a workload that consists of \( N+1 \) jobs consisting of \( N \) type A jobs and one type B job being serviced by the simple scheduling scheme described below. Job type A has a CPU demand of \( a \) seconds, and job type B has a demand of \( b \) seconds. Furthermore, \( b \) is a large integer multiple of \( a \), and type A jobs arrive at fixed time points \( X, 2X, 3X, \) etc. where \( X \) is a positive integer multiple of \( a \); i.e., \( X = ka \) where \( k \) is a positive integer.

In this system, type A jobs have a higher priority than the one type B job and will preempt (with 0 overhead) any type B job from the CPU.

a) Draw the space-time diagram (time runs to the right) for the case when \( a = 1, b = 10, k = 2, \) and \( N = 4 \).

b) Derive an expression for the turnaround time and the queueing time of the type B job and each type A job. Here, you need to handle the general case of arbitrary \( N \) and \( k \) although subject to the constraints specified earlier; i.e., \( k \) is a positive integer, and \( b \) is a large integer multiple of \( a \).
Problem 4 (4 Points) [From Tanenbaum (modified)]

Suppose that a machine has 48-bit virtual addresses with a single-level page table and 32-bit physical addresses. Pages are 4 KB.

a) How many bits should be allocated for each of the page number and the offset fields in the virtual (logical) address? Explain.

b) How many entries are in a single-level page table?

c) How many pages would be needed for the page table in Part a if each entry occupies 4 bytes?

d) If the program and data together fit in page 0 and the stack fits in the highest page, how many page table entries are needed for two-level paging if an equal number of bits are used to represent each part of the page number? Explain.

Problem 5 (6 Points)

Modify the ucontext-basic.c program so that it creates 4 instances of the childFiber thread that executes the algorithm shown below. The details are the following:

- The id of kth childFiber is k.
- The algorithm for each childFiber is:

  Display "BEGIN" followed by my id;
  Do 2 times {
    Sleep for 1 second;
    Yield to my neighbor thread;
    Display "RESUME" followed by my id;
  }
  Sleep for 1 second;
  If I am not thread 3 Then
    Yield to my neighbor thread;
  Else
    Yield to the main thread;
  End
  Display "END" followed by my id;

In the above algorithm, thread \((k + 1) \ mod 4\) is the next neighbor of thread \(k\); i.e., the threads form a ring control structure. The id is the instance number; i.e., 0, 1, 2, or 3.

Submit the following:

a) Your source listing

b) The output of your program

c) A short explanation of why your output shows that your program is functioning properly.