Department of Computer Science
University College London

1B11 Operating Systems Coursework
December 2000

1. Assume that there is a system call on your operating system that allows you to read or write sector N on the disk. Disk transfers are carried out using DMA and an interrupt is raised when the operation completes. Explain clearly in your own words, using any diagrams you think will help, exactly what happens in the Operating System between a user application making a system call to read sector 50 from disk and the data being returned to the user's buffer. You should indicate any process scheduling that occurs, changes in processor state, memory mapping changes, the use of interrupt service routines and anything else you think is relevant.

2. Processes with the indicated CPU requirements are queued in an operating system awaiting execution (i.e. there is a backlog and no new processes are being added to the queue).

Job Number	CPU Time
1	200
2	600
3	100
4	300
5	500
6	400
7	300
8	700
9	300
10	600

Assume that it takes 50 time units for context switching. For each of the scheduling algorithms below:

i.   Calculate the time to completion of each process (TTC),
ii. Calculate the average TTC
iii. Calculate the efficiency, where Efficiency = useful work/total work.
iv. Indicate the main benefit of each algorithm.

The scheduling algorithms are:

a) Shortest job first, run to completion (SJF, RTC)
b) First come first served, run to completion (FCFS, RTC)
c) First come first served, time sliced (150 unit slice) (FCFS, TS 150)
d) First come first served, time sliced (250 unit slice) (FCFS, TS 250)
e) Shortest remaining time, time sliced (250 unit slice) (SRT, TS 250)

Please Turn Over

Which of these algorithms would be preferable for scheduling a multi-user processor? Why?

Which of these algorithms would be preferable for scheduling a multi-user print server? Why?

Hint:

You may find it easiest to tabulate the progress of processes. For Run to Completion scheduling a table of the form below might be used.

Job Number	CPU Time	TTC

For Time Sliced scheduling the following table structure may be useful:

Job	CPU T	T1	Rem 1	T2	Rem 2	T3	Rem 3	T 4	TTC	C #

Where:

Job	*is the job number*	*CPU T*	*is the CPU time*
T1 … T4	*are the time taken by the job in each of the scheduling cycles*	*Rem1 … Rem3*	*is the remaining time for each job after the indicated scheduling cycle*
TTC	*is the time to complete the job*	*C #*	*is the order in which the jobs complete (1 = first, etc.)*

PLEASE NOTE THAT THIS IS GRADED COURSEWORK. A STUDENT WHO FAILS TO COMPLETE IT WILL BE AWARDED ZERO MARKS FOR IT.

THIS COURSEWORK SHOULD BE HANDED IN TO THE DEPARTMENTAL OFFICE NO LATER THAN:

12 noon on Friday 12th January 2001.

THERE ARE FIXED PENALTIES FOR HANDING YOUR WORK IN LATE -- THEY ARE:

<= 2 WORKING DAYS   10% OF MARK OBTAINED

> 2 WORKING DAYS   100% OF MARK OBTAINED