UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2001

BEng Honours Degree in Computing Part I
MEng Honours Degrees in Computing Part I
for Internal Students of the Imperial College of Science, Technology and Medicine

This paper is also taken for the relevant examinations for the Associateship of the City and Guilds of London Institute

PAPER C111

OPERATING SYSTEMS I

Tuesday 1 May 2001, 14:00
Duration: 90 minutes
(Reading time 5 minutes)

Answer THREE questions

Paper contains 4 questions
Calculators required

1a List briefly the steps required to assemble a user's assembly-language program so that it is ready to be executed from the main store of a computer. The store is initially empty, but can be loaded by means of switches on the computer's control panel. You may assume that the following tapes are available:

i a tape containing a *relocating* loader in absolute binary format.

ii a tape containing a *two-pass* assembler in relocatable binary format.

iii a tape containing the user's assembly-language program in source form.

iv a tape containing a *linking loader* in relocatable binary format.

v a library tape containing standard subroutines in object code format.

Show the state of the store after each step by means of diagrams.

b *Briefly* describe what is meant by *spooling system* and list the uses to which it may be put. *Do not* describe its implementation.

The two parts carry, respectively, 70%, 30% of the marks.

2 An operating system comprises several terminal processes, a loader, a dispatcher and a resource manager. Each terminal process has a small amount of store permanently allocated to it. The remaining free store is divided into several partitions which can be allocated to terminal processes under the control of the resource manager.

a Describe the sequence of events which occurs when a terminal process which does not initially have a partition has to load and run a program for a user. The partition is dedicated solely to the terminal process until the loaded program has finished. Indicate clearly the transfers of control between the different system components and state what information is passed between them.

b The designers wish to speed up the system by allowing other terminal processes to *reuse* a partition when the program currently loaded in it is waiting for terminal input. Describe the extra actions which must be taken in addition to those in (a) above to make this possible. Identify clearly any additional components (both code and data) which must be added to the system to support this new facility.

The two parts carry, respectively, 50%, 50% of the marks.

3 A certain multiprocessing system runs `ProcessCount` concurrent processes, *any* of which may request an input-output (i/o) operation by branching to a standard i/o package. This initiates the required operation, copies the address of the device performing it into a global variable `DeviceUsed` of type `DeviceAddress` and *waits* by branching to the dispatcher. The branch mechanism causes a return address to be placed in a global variable `ReturnAddress` of type `StoreAddress`. The dispatcher maintains a data structure whose declaration is as follows:

```
type WaitFor : enum ( Start , IO )
type ProcessTable : array 1 .. ProcessCount of
                    record
                    ProcessState  : WaitFor
                    StartAddress,
                    ResumeAddress : StoreAddress
                    ActiveDevice  : DeviceAddress
                    end record
```

a Using a Turing-like language, write a dispatcher which gives each process an *equal chance* of running and which updates a global variable `ActiveProcess` of type `1 .. ProcessCount` which contains the number of the currently active process. The following functions and procedures are available to you:

```
function SystemRunning : boolean
% Returns true after the console STOP button has been pressed; no more
processes are to be dispatched after the currently active process requests i/o.
function DeviceReady ( Device : DeviceAddress ) : boolean
% Interrogates Device, returning true if its i/o operation is complete.
procedure JumpTo ( Location : StoreAddress )
% Transfers control to the instruction held in Location.
```

b Suggest what new procedures should be added to the set given above in order to allow processes in the system to wait for *other processes* as well as for i/o. Explain *clearly* what each one is used for, but *do not* show its internal details. Show the modifications which would be needed to the code of the dispatcher and to the data structure `ProcessTable` to implement this facility using the new procedures.

The two parts carry, respectively, 50%, 50% of the marks.

4a Show by means of diagrams the *physical* organisation of each of the following types of filing system:

i For a system with *one* disk, in which each file occupies a *single area* on the disk. List the advantages and disadvantages of such a system.

ii For a system with *one* disk in which each file may occupy *more than one* area of disk. Which disadvantages of system (a) have been overcome in this system? List any new disadvantages.

b Show by means of diagrams the *logical* organisation of a *hierarchical* filing system suitable for a system with *more than one* disk. List the advantages and disadvantages of such a system over one which has a single directory for all disks.

The two parts carry, respectively, 70%, 30% of the marks.