CMR

INSTITUTE OF TECHNOLOGY

SCHEME AND SOLUTION

Improvement test – May 2017

Sub:	OPERATING SYSTEMS					**Code:**10EC65		
Date:	30/ 05 /2017	Duration:	90 mins	Max Marks:	50	**Sem:**	VI A & B	**Branch:ECE**

Note: Answer any five questions:

1. Expalin the user level,kernel level and hybrid thread models with neat diagrams — 10M

Explanation about kernel level thread with diagram:3M
Explanation about user level thread with diagram:3M
Explanation about Hybrid level thread with diagram:4M

- Kernel-Level Threads
  - Threads are managed by the kernel
- User-Level Threads
  - Threads are managed by thread library
- Hybrid Threads
  - Combination of kernel-level and user-level threads

Kernel-Level Threads

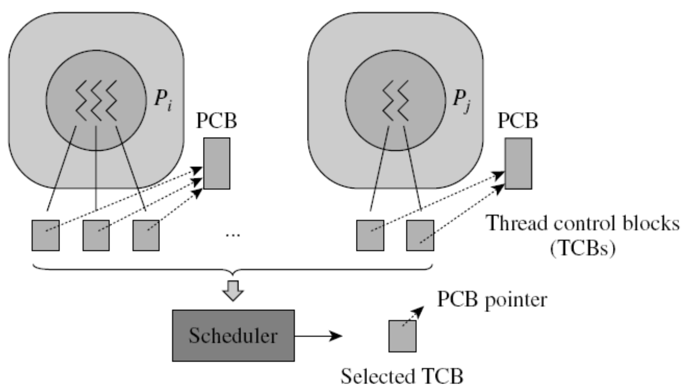

Figure 5.14 Scheduling of kernel-level threads.

- A kernel-level thread is like a process except that it has a smaller amount of state information
- Switching between threads of same process incurs the overhead of event handling

User-Level Threads

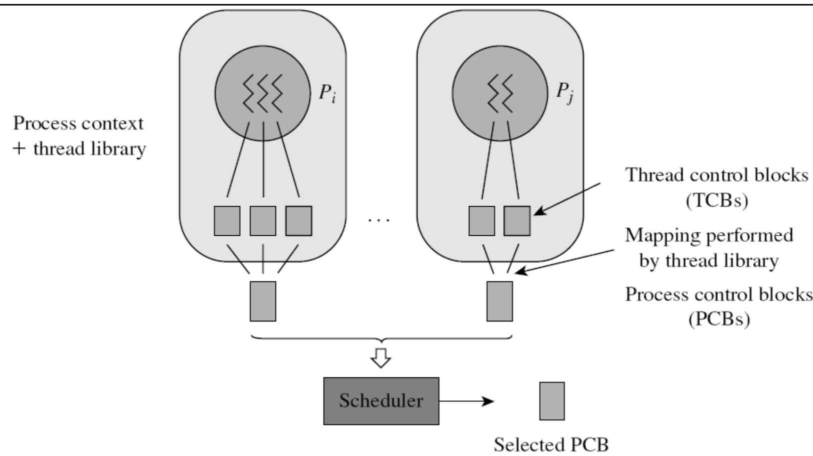

Figure 5.15 Scheduling of user-level threads.

- Fast thread switching because kernel is not involved
- Blocking of a thread blocks all threads of the process
- Threads of a process: No concurrency or parallelism

Hybrid Thread Models

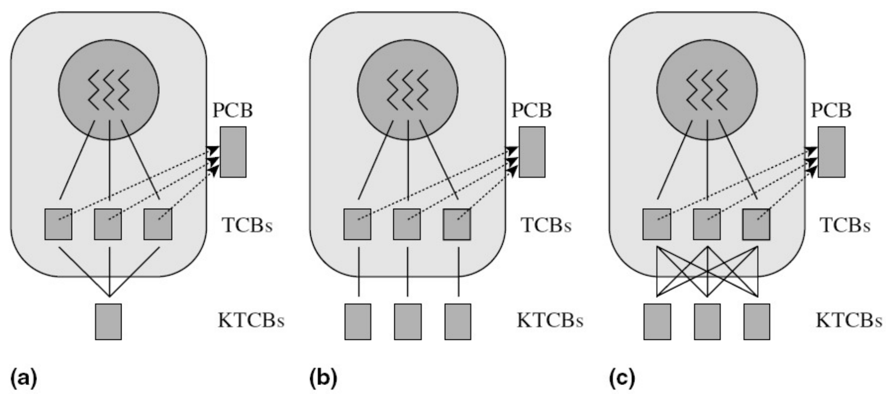

Figure 5.17 (a) Many-to-one; (b) one-to-one; (c) many-to-many associations in hybrid threads.

- Can provide a combination of parallelism and low overhead

2. With a neat diagram explain threads in Solaris. Give the process state diagram in Unix operating systems. — 10M

Explanation about the in solaris with diagram- 7M

Diagram of process state for Unix OS-3M

- Three kinds of entities govern concurrency and parallelism within a process:
  - User threads
  - Lightweight processes (LWPs)
    - Provides arallelism within a process
    - User thread are mapped into LWPs
  - Kernel threads
- Supported two different thread models
  - M x N model upto solaris 8
  - 1 : 1 model Solaris 8 onwards
- Provides scheduler activations to avoid thread blocking and notify events

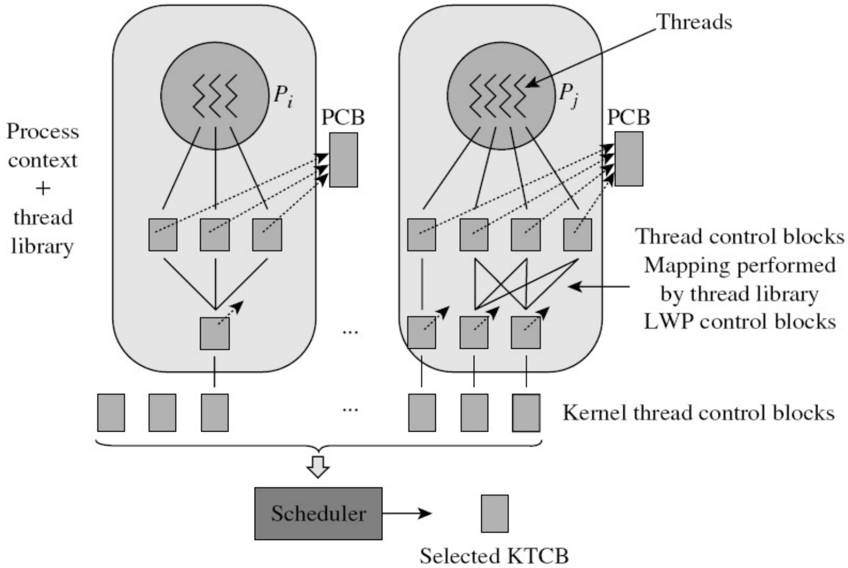

Figure 5.21 Threads in Solaris.

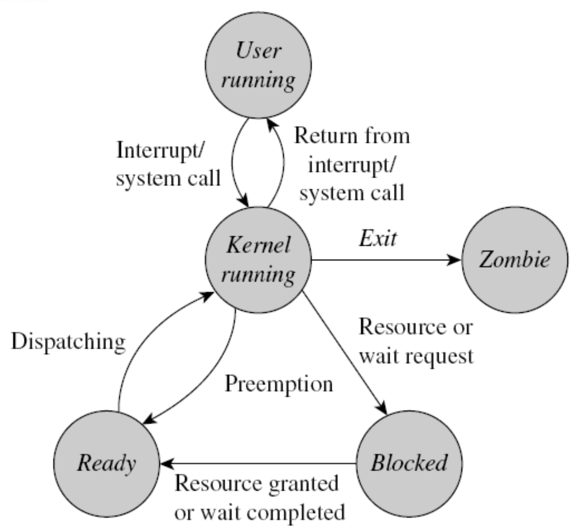

Figure 5.19 Process state transitions in Unix.

3. (a)Explain the interface between File system and Input Output control System(IOCS). 7M

Explanation-3M
Diagram-4M

- Interface between file system and IOCS consists of
  - *File map table* (FMT)
  - *Open files table* (OFT)
  - *File control block* (FCB)

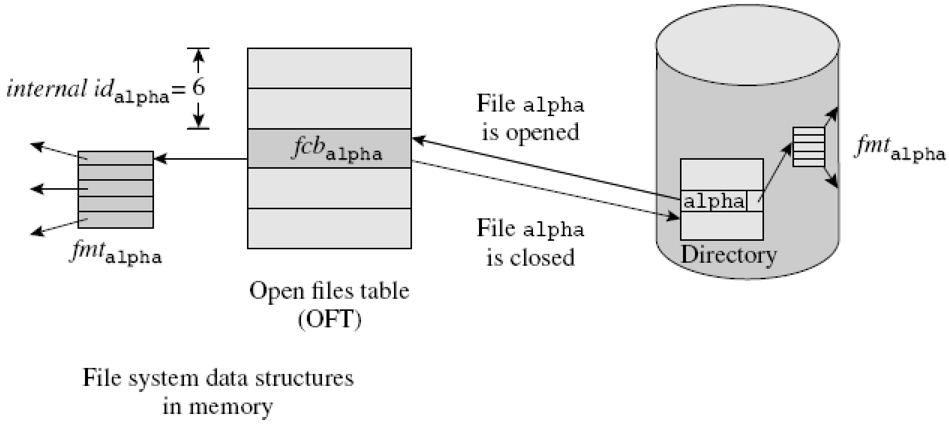

Figure 13.18 Interface between file system and IOCS—OFT, FCB and FMT.

Table 13.3 Fields in the File Control Block (FCB)

Category	Fields
File organization	File name File type, organization, and access method Device type and address Size of a record Size of a block Number of buffers Name of access method
Directory information	Information about the file's directory entry Address of parent directory's FCB Address of the file map table (FMT) (or the file map table itself) Protection information
Current state of processing	Address of the next record to be processed Addresses of buffers

b)Expalin the opeartions performed on Files. 3M

Defining at least 6 operations-3M

Table 13.2 Operations on Files

Operation	Description
Opening a file	The file system finds the directory entry of the file and checks whether the user whose process is trying to open the file has the necessary access privileges for the file. It then performs some housekeeping actions to initiate processing of the file.
Reading or writing a record	The file system considers the organization of the file (see Section 13.3) and implements the read/write operation in an appropriate manner.
Closing a file	The file size information in the file's directory entry is updated.
Making a copy of a file	A copy of the file is made, a new directory entry is created for the copy and its name, size, location, and protection information is recorded in the entry.
File deletion	The directory entry of the file is deleted and the disk area occupied by it is freed.
File renaming	The new name is recorded in the directory entry of the file.
Specifying access privileges	The protection information in the file's directory entry is updated.

4. With the help of diagrams,discuss Index sequential file organization with an example — 10M

Explaning the concept-4M
3 diagrams-6M-2marks each

- An *index* helps determine location of a record from its key value
  - Pure indexed organization: (key value, disk address)
  - *Index sequential* organization uses index to identify section of disk surface that may contain the record
    - Records in the section are then searched sequentially

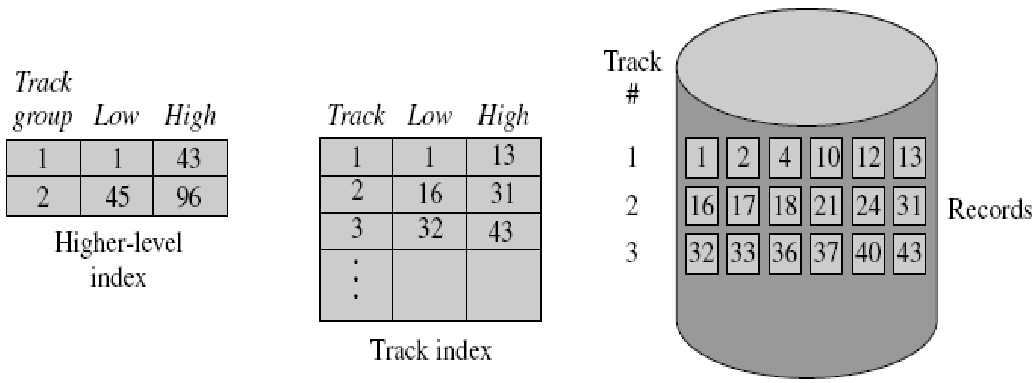

Figure 13.5 Track index and higher-level index in an index sequential file.

5. Determine mean turnaround time and mean weighted turnaround time for following set of processes with appropriate graphs **Least Completed Next(LCN and Shortest Time to Go (STG**) scheduling algorithms. — 10M

Processes	P_1	P_2	P_3	P_4	P_5
Admission time	0	2	3	4	8
Service Time	3	3	5	2	3

Calculating mean turnaround time and mean weighted turnaround time for LCN-5M
Calculating mean turnaround time and mean weighted turnaround time for STG-5M

Time of scheduling		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
CPU time consumed by processes	P_1	0	1	2	2	2	2	2	2	2	2	2					
	P_2			0	1	1	1	2	2	2	2	2	2				
	P_3				0	1	1	1	2	2	2	2	2	2	3	4	5
	P_4					0	1	1	1								
	P_5									0	1	2	2	2	2		
Process scheduled		P_1	P_1	P_2	P_3	P_4	P_2	P_3	P_4	P_5	P_5	P_1	P_2	P_3	P_5	P_3	P_3

c	t_a	w
11	11	3.67
12	10	3.33
16	13	2.60
8	4	2.00
14	6	2.00

$\overline{t_a}$ = 8.8 seconds, $\overline{w}$ = 2.72

c: completion time of a process

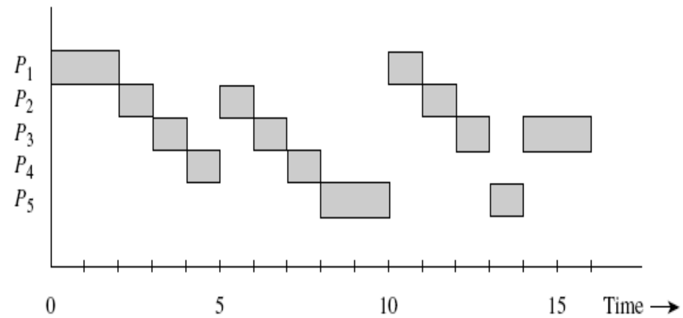

Figure 7.7 Scheduling using the least completed next (LCN) policy.

Mean turnound time ta=8.8s
Mean weighted turnound time=2.72

Shortest Time to Go (STG)

Time of scheduling		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Remaining CPU time requirement of a process	P_1	3	2	1													
	P_2			3	3	2	2	2	1								
	P_3				5	5	5	5	5	5	5	5	5	4	3	2	1
	P_4					2	1										
	P_5									3	2	1					
Process scheduled		P_1	P_1	P_1	P_2	P_4	P_4	P_2	P_2	P_5	P_5	P_5	P_3	P_3	P_3	P_3	P_3

c	t_a	w
3	3	1.00
8	6	2.00
16	13	2.60
6	2	1.00
11	3	1.00

$\overline{t_a}$ = 5.4 seconds, $\overline{w}$ = 1.52

c: completion time of a process

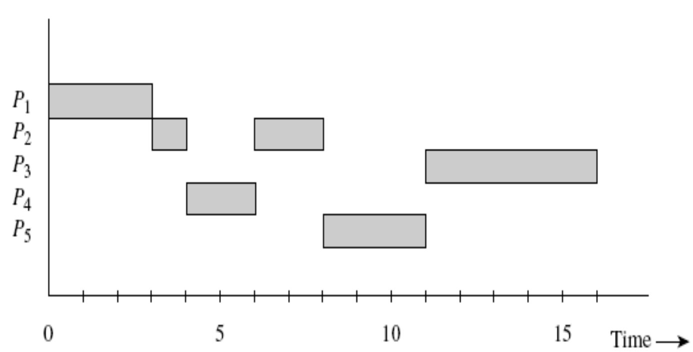

Figure 7.8 Scheduling using the shortest time to go (STG) policy.

Mean turnound time ta=5.4s
Mean weighted turnound time=1.52

6. With a neat diagram, explain the facilities provided by file system and IOCS layers. 10M

Explanation about the file system with IOCS with diagram-5M
Describing the facilities-5M

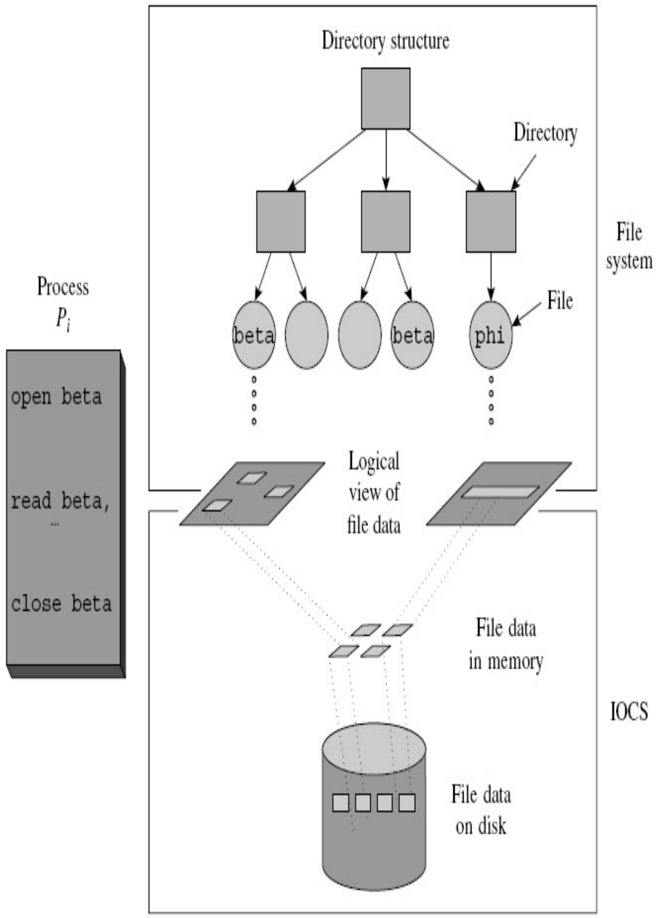

Figure 13.1 File system and the IOCS.

- File system views a file as a collection of data that is *owned* by a user, *shared* by a set of authorized users, and *reliably stored* over an extended period
- IOCS views it as a repository of data that is *accessed speedily* and stored on I/O device that is *used efficiently*

Table 13.1 Facilities Provided by the File System and the Input-Output Control System

File System
- Directory structures for convenient grouping of files
- Protection of files against illegal accesses
- File sharing semantics for concurrent accesses to a file
- Reliable storage of files

Input-Output Control System (IOCS)
- Efficient operation of I/O devices
- Efficient access to data in a file

- Two kinds of data: *file data* and *control data*

7. (a) What is a link? with an example ,illustrate the use of a link in an acyclic graph structures — 4M

Definition-1M

Explanation with graph-3M

- Tree structure leads to a fundamental asymmetry in the way different users can access a shared file
  - Solution: use *acyclic graph structure* for directories
    - A *link* is a directed connection between two existing files in the directory structure

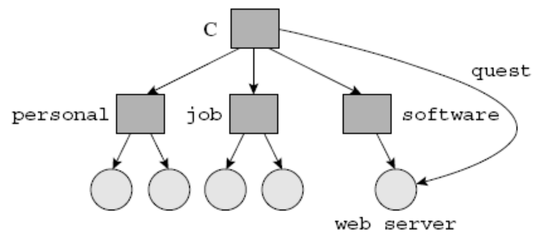

Figure 13.9 A link in the directory structure.

(b) With an example explain sequential and direct file access methods. Describe the operation of real time process using Rate Monotonic Scheduling(RMS) for the following three independent processes that do not perform I/O operations . — 6M

Process	P_1	P_2	P_3
Time period(ms)	10	15	30
Service Time(ms)	3	5	9

Diagram for sequential and direct file access methods-1mark each-1*1=2M

Describing the operation of Monotonic Scheduling(RMS) with graph-4M

- Determines the *rate* at which process has to repeat
  - Rate of $P_i = 1 / T_i$
- Assigns the rate itself as the priority of the process
  - A process with a smaller period has a higher priority
- Employs a priority-based scheduling
    - Can complete its operation early

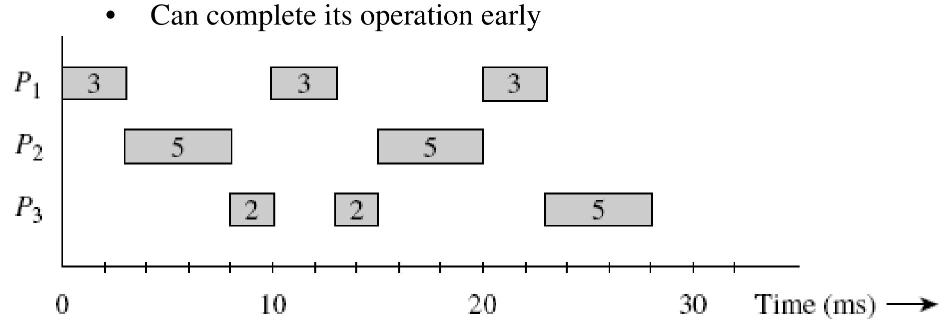

Figure 7.14 Operation of real-time processes using rate monotonic scheduling.

8	Explain unix virtual memory using paging Explanation-10M • Paging hardware differs in architectures • Pages can be: resident, unaccessed, swapped-out • Allocation of as little swap space as possible • Copy-on-write for *fork* • Lack reference bit in some HW architectures; compensated using *valid bit* in interesting manner • Process can fix some pages in memory • *Pageout daemon* uses a clock algorithm – Swaps out a process if all required pages cannot be in memory A swap-in priority is used to avoid starvation	10M