1

24/05/21

END SEM EXAMS
OPERATING SYSTEMS

JUNAID . GIRKAR
G00041900057
SE COMS A-3

Q1 A What is Process Control Block ? With the help of diagram, explain all the elements of PCB

ANS
- Any process is identified by its Process Control Block (PCB) which is the data structure used by the operating system to keep track on the processes .
- All the information associated with the process is kept in PCB , there is seperate PCB for each process.
- Traffic controller module keeps track on the status of the processes
- If many processes wait for the same device then PCB's of all these processes are linked together in chain waiting for that device.
- If device becomes free then traffic controller checks the PCB chain to see if any process is waiting for the device .
- If the processes are available in the device chain, then again it is placed back to ready state to request that device again.

IDENTIFIER
STATE
PRIORITY
PROGRAM COUNTER
MEMORY POINTERS
I/O STATUS INFORMATION
ACCOUNTING INFORMATION
⋮

A simplified PCB

6000u190057

IDENTIFIER : A unique identifier associated with this process to distinguish it from all other processes

STATE : If the process is currently executing, it is in the running state

PRIORITY : Priority level relative to other processes

PROGRAM COUNTER : The address of the next instruction in the program to be executed.

MEMORY POINTERS : Includes pointers to the program code and data associated with this process plus any memory blocks shared with other processes

CONTEXT DATA : These are data that are present in registers in the processor while the process is executing

I/O STATUS INFORMATION : Includes outstanding I/O request, I/O devices assigned to this process, a list of files in use by the process and so on.

ACCOUNTING INFORMATION : May include the number of processor time and clock time used, time limits, account numbers and so on.

6000419005J

Q1 B Explain different functions of operating systems

ANS The operating system performs the following functions in a device:-

- ~~Instruction~~
- ~~I/O management~~

1) MEMORY MANAGEMENT: It is the management of the main memory. whatever program is executed, it has to be ~~executed~~ present in the main memory. Therefore, there can be more than one program present at a time. Hence OS is required to manage the memory by allocation, deallocation, distribution and recording.

2) PROCESSOR MANAGEMENT: when more than one process runs on the system, the OS decides how and when a process will use the CPU. Hence the name is also CPU scheduling where the OS allocates, deallocates processors and keeps record of CPU status. It uses certain CPU scheduling algorithms like FCFS, SJF, Priority, Round-Robin, etc.

3) DEVICE MANAGEMENT: The processes may require devices for their use. The OS decides the need of the processes and allocates/deallocates devices accordingly.

4) FILE MANAGEMENT: The files on a system are stored in different directories. The OS keeps record of the status and location of files and allocates/deallocates resources.

5) SECURITY: The OS keeps the system and programs safe and secure through different methods of authentication

6) ACCOUNTING

7) ERROR DETECTION

8) RECORDS OF SYSTEM PERFORMANCE

9) COMMUNICATION BETWEEN DIFFERENT SOFTWARES, etc.

B0004190057

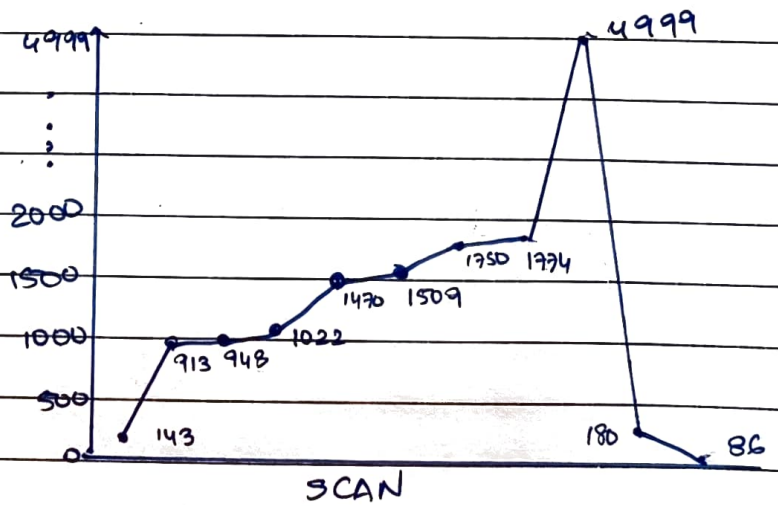
Q3

5000 cylinders [0 - 4999]
current request = 143
Previous request = 125

Queue : 86, 1470, 913, 1774, 948, 1509, 1022, 1750, 130

(i) The SCAN schedule is 143, 913, 948, 1022, 1470, 1509, 1750, 1774

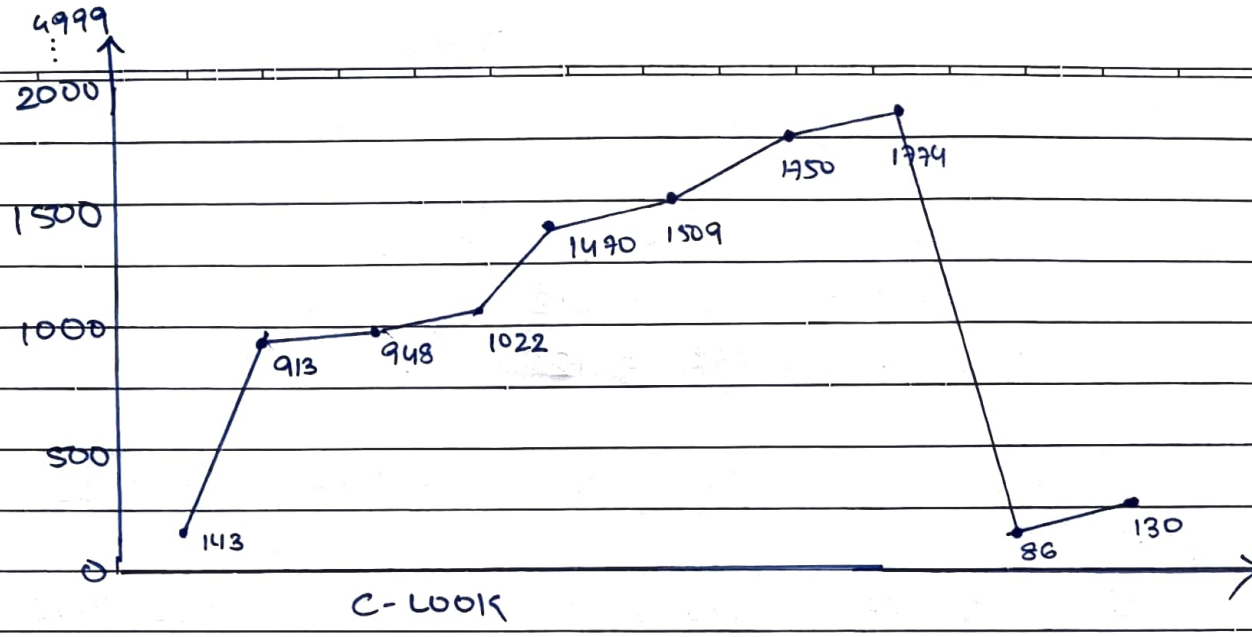
Total distance travelled = |143 - 913| + |913 - 948| + |948 - 1022|
+ |1022 - 1470| + |1470 - 1509| + |1509 - 1750|
+ |1750 - 1774| + |1774 - 4999| + |4999 - 180| + |180 - 86|
= 9769



(ii) The C-LOOK schedule is 143, 913, 948, 1022, 1470, 1509, 1750, 1774, 86, 130.

Total distance travelled = |143 - 913| + |913 - 948| + |948 - 1022| + |1022 - 1470|
+ |1470 - 1509| + |1509 - 1750| + |1750 - 1774| + |1774 - 86| + |86 - 130|
= 3363

6000U190057



C-LOOK

B00041900 57

Q.5

	ALLOCATION				MAX				AVAILABLE			
	A	B	C	D	A	B	C	D	A	B	C	D
P_0	0	0	1	2	0	0	1	2	1	5	2	0
P_1	1	0	0	0	1	7	5	0				
P_2	1	3	5	4	2	3	5	6				
P_3	0	6	3	2	0	6	5	2				
P_4	0	0	1	4	0	6	5	6				

ANS i) Need matrix = Max − Allocation

=

	NEED			
	A	B	C	D
P_0	0	0	0	0
P_1	0	7	5	0
P_2	1	0	0	2
P_3	0	0	2	0
P_4	0	6	4	2

ii) To decide whether system is in safe state or not.

PROCESS P0 : Need <= Availability [(0,0,0) <= (1,5,2,0)] is True

P0 is executed.

New Availability = Availability + Allocation

= (1,5,2,0) + (0,0,1,2)

= (1,5,3,2).

PROCESS P1 : Need <= Availability [(0,7,5,0) <= (1,5,3,2)] is False

∴ P1 is not executed.

PROCESS P2 : Need <= Availability [(1,0,0,2) <=(1,5,3,2)] is True

∴ P2 is executed.

New Availability = Availability + Allocation

= (1,5,3,2) + (1,3,5,4)

= (2,8,8,6)

PROCESS P3 : Need <= Availability [(0,0,2,0) <= (2,8,8,6)] is True

∴ P3 is executed

New availability = Availability + Allocation

= (2,8,8,6) + (0,6,3,2)

= (2,14,11,8)

PROCESS P4 : Need <= Availability [(0,6,4,2) <= (2,14,11,8)] is True

∴ P4 is executed

New availability = Availability + Allocation

= (2,14,11,8) + (0,0,1,4)

= (2,14,12,12)

PROCESS P1 : Need <= Availability [(0,7,5,0) <= (2,14,12,12)] is True

∴ P1 is executed.

New Availability = Availability + Allocation

= (2,14,12,12) + (1,0,0,0)

= (3,14,12,12)

∴ SYSTEM IS IN SAFE STATE [P0, P2, P3, P4, P1]

(iii) Safe sequences :- [P0 → P2 → P1 → P3 → P4] ;

[P0 → P2 → P3 → P1 → P4] ; [P0 → P2 → P4 → P1 → P3] ;

[P0 → P2 → P1 → P4 → P3] ; [P0 → P2 → P4 → P3 → P1] ;

[P2 → P3 → P4 → P0 → P1] ; [P3 → P4 → P0 → P2 → P1] .

5

8000 419005 7

Q6 Explain Paging and segmentation w.r.t virtual memory

	PAGING	SEGMENTATION
•	Main memory partitioned into small fixed-size chunks called frames	Main memory not partitioned
•	Program broken into pages by the compiler or memory management system.	Program segments specified by the programmer to the compiler.
•	Internal fragmentation within frames	No internal fragmentation
•	No external fragmentation	External fragmentation
•	OS must maintain a page table for each process showing which frame each page occupies	OS must maintain a segment table for each process showing the load address and length of each segment
•	OS must maintain a free frame list.	OS must ~~main a tis~~ maintain a list of free holes in memory
•	Processor uses page number, offset to calculate absolute address	Processor use segment number, offset to calculate absolute address

• Not all pages of a process need to be in main memory frames for the process to run. Pages may be read in as needed.	Not all segments of a process need be in main memory for the process to run. segments may be read in as needed
• Reading a page into main memory may require writing a page out to disk	Reading a segment into main memory may require writing one or more segments out to disk.

B000419005 7

Q4 a] Real time OS

b] Short term scheduler, Medium term scheduler, long term scheduler.

ANS a] Real Time operating systems [RTOS] are used in environments where a large number of events, mostly external to the computer system, must be accepted and processed in a short time or within deadlines. Such applications are industrial control, telephone switching equipment, flight control and real time simulations.

RTOS can be of 2 types :-

HARD RTOS : These operating systems guarantee that critical tasks be completed within a range of time.

For e.g : A robot is hired to weld a car body, if the robot welds too early or late, the car cannot be sold, so a hard RTOS is used.

SOFT RTOS : This operating system provides some relaxation in time limit.

For e.g : Multimedia systems, digital audio systems, etc. Explicit, programmer defined and controlled processes are encountered in real time systems. A seperate process is changed with handling a single external event. The process is activated upon occurance of the related event signaled by an interrupt.

Priority based preemptive scheduling is used by RTOS

Advantages :-
- Maximum consumption
- Task shifting
- Error free
- Focus on Application

Disadvantages :-
- Limited tasks
- Use heavy system resources
- Complex algorithms
- Thread priority.

ANS b] LONG TERM SCHEDULER :-
- When programs are submitted to the system for the purpose of scheduling, long term scheduler's job is to choose processes from the queue and place them into main memory for execution purpose
- Long-term scheduler controls the degree of multiprogramming
- It is not present in time shared OS

SHORT TERM SCHEDULER :-
- Process which are in ready queue wait for CPU. A short term scheduler chooses the process from ready queue and assigns it to the CPU based on some policy.
- Short term scheduler is faster than long term scheduler.

8000419005?

MEDIUM TERM SCHEDULER

- If the degree of the multiprgramming increases, medium-term scheduler swap out the processes from main memory.
- The swapped out processes are again swapped in by the medium-term scheduler
- This is done to control the degree of multiprogramming or to free up memory.