28/1/2020

Date / /
Page

Operating System :- It is an organized collection of program to provide an extension for hardware. it consist of control routin for operating a computer system and provide communication link between user and hardware. it is a system software that x as an intermedialy agent between user and computer hardware. It provides working environment to execute programs in convinion and efficient manner.) It is responsible of overviewing of functioning of machine from booting to resource management.

1. Managing all the resources. (devices)
2. Providing interface between user and system.
29-1-2020 3. Managing the memory allocation and reallocation to the process.
4. Scheduling the computational process.
5. Managing the files and disk.
6. Protection ~~the~~ of system and data.
7. Resource allocation and management.
8. Sharing of resource in term of time space and C.P.U to achieve optimal utilization of resources.

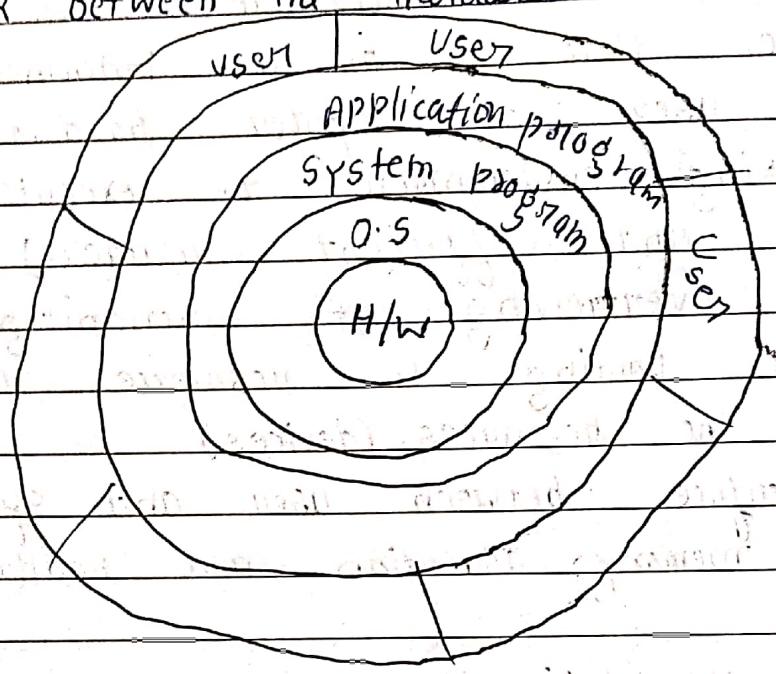
Op. Objective system object :- The primary objectives of s are -

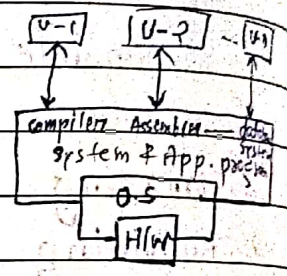
1. To increase productivity of computer resources.
2. Facilitate the use of hardware.
3. Optimal utilization of computing resources.
4. Scheduling computational activites to ensure good performance of system.
5. Creation of user friendly environment for developmen of and execution of program.

Scanned with CamScanner

Date / /
Page

O.S and its basic structure :- O.S is an essential part of system which directly control computer hardware resources. it facilitates the user application to make the communication link between the hardware.





Functions of O.S :- Fundamentally n O.S performs following functions.

1. Transfer i/o from the Key board to the memory.
2. Stored data or program in external storage devices.
3. Control the peripherals.
4. load programs and packages from storage devices and media to the main memory.
5. Execute user programs and commands.
6. Provide security and protection to the user program and file.

The basic Functional block of an O.S :-

1. Processor management and scheduling :- O.S manages peripherals and resource of the computer. Instructions

Date / /
Page

and commands and schedule the job for the C.P.U. It also handles sudden interupts and schedule them accordingly.

2. device management :- There are two ~~bits~~ devi ~~of dev~~ points of device connected to the computer.
(i) Sharable device :- These device enables many user symontanious ly to main memory ~~and~~ or disk.
(ii) Non sharable device :- These device can ~~be~~ ~~used~~ be used by one device at a time (Keyboard, monitor). The O.S monitors the availablity and status of these devices.

3. File and data management :- It managing user programs and data files. O.S performs following activites.
(i) set up directries and stores user file.
(ii) Transfer M.M contains in specific file name to disk and viceversa.

4. Memory management :- The O.S partition the memory for different users and programs in multi user system. memory allocation is dynamic. The avalible of memory for processing different activites is manage by the O.S.

5. Command interpretation and execution :- O.S reads the commands enter by the user checks for errors interpreates and translate them to machine Instruction and gets executed by the Hardware.

Scanned with CamScanner

6. Job accounting and log maintainance :- O.S with the help of an internal time clock computers. The time required by the each resource for different users. It also maintaince a log. and keep tract of login or logout time.

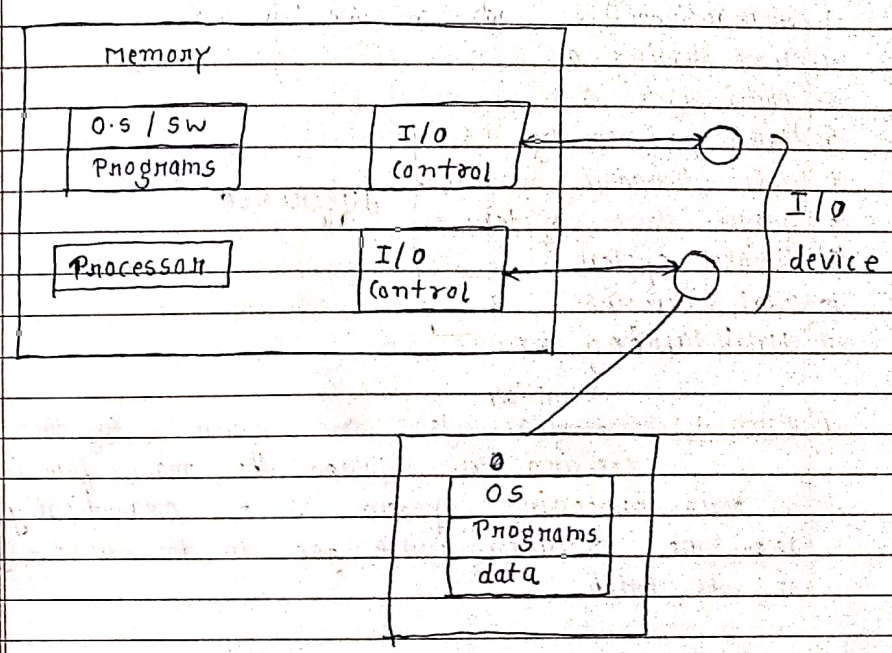
7. Communication :- It is the facility provided to user to communicate with each other in the same time by sharing or transfering data.

8. Fault tolerance ~~of~~ and management :- ~~E~~ It is built in with O.S providing replication (duplication of a data) in the event of any interface or peripheral device failure. The system automatically switch over to the substitute.

-2020

Role of O.S :- An extended virtual machine to intract or gain the access of computer system it works as virtual machine and provides the way of communication it controls the service of invironment.

2. A resource manager :- it manages all the computer source for optimal utilization and speed. it usually manges process, memory, file, i/o management etc. it ~~tel~~ also helps in C.P.U. utilization and security protection of system ~~and~~ or data.



3. A service provider :- it provides the services to intract with application program for applicability for each and every application of system software or O.S is responsible.

4. A security manages :- it is responsible for system and data software. it controls the access of service resources and data as a security agent.

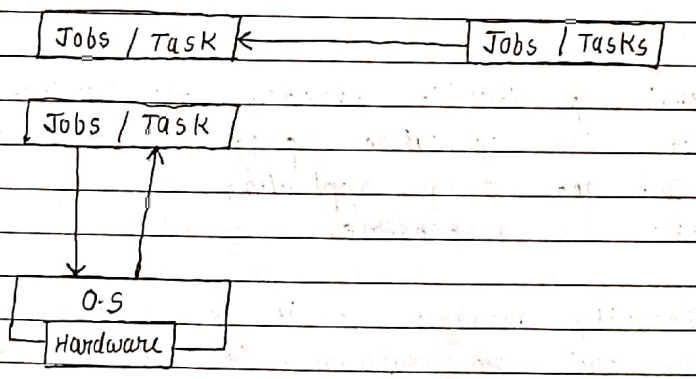
Classification of O.S :-

1. Batch O.S
2. Intractive O.S

3. Distributive O.S
4. Time sharing O.S
5. Multi user O.S
6. Multi programming O.S
7. Multi threaded system
8. Single user system
9. Parallel system
10. Real time O.S
11. Multi tasking O.S

difference

1. Batch O.S :- ~~Singular~~ Similiar jobs are batched together and executed in groups. The main function of a batch processing system is to automatically keep the executing one job to the next job in the batch.



2. The main idea behind a batch processing system is to reduce the interference of the operator during the processing of job by the compider.
3. All functions of a batch processing system are carried out by batch monitor.
3. In batch processing system the turn around time is defined as the time from which a user job is given to the time when its output is given back to the user.

a ← b
happened before

DDOS (distributive denia of services)

4. Process scheduling memory management and I/O management is quite simple.
5. Jobs are proceeds in order in summions as FCFS. (first come first serve)
6. Memory is divided in two sections: one fix section to contain O.S routins and other for user programs.
7. Batch O.S can be single program system and multi program system.

Advantage :-

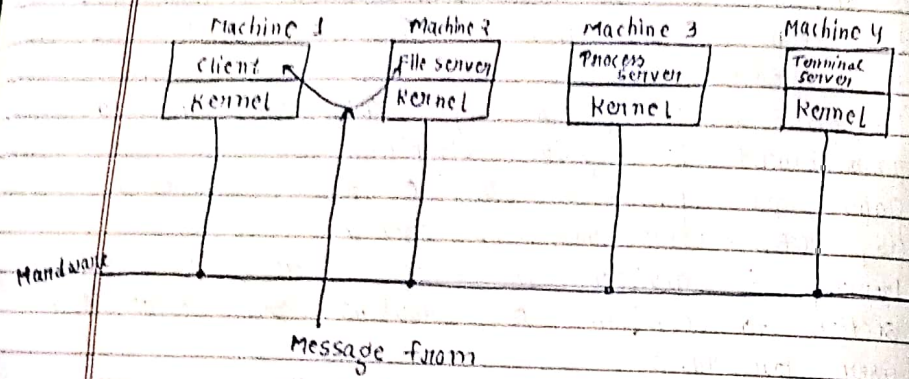
1. Move much of the work of the operator to the computer.
2. Increased performance since it was processible a job to start as soon as the previous job finished.

Disadvantage :-

1. Known interactive environment: There is no interaction between user and program.
2. Offline debbiging :- bugs can't be connected at the moment they occur.

31-1-2020

2. Distributed O.S :- A recent trend in computer system is to distribute amputation among several processors.



Each processor has its own memory and clock. The processor in distributed system may very insize and function. They may include small micro processors Each work station mini computer and large general purpose computer system. Distributed O.S resources in distributed system shemelessly and transparently to the user. it looks to the user like a centralize O.S but operates on multiple independent C.P.U. it can be an extension of client server concept where the server are remote.

Advantages :-

1. Incremental growth :- computer power can be added in small increments.
2. Liblity :- if one machine crashes the system as a whole can still survive.
3. Speed :- A distributed system may have more total computer power than a main frame.

TCP/IP VOP VPN Virtual private network

4. Open less :- it is an open system. it is already to communicate with other system.
5. Economy :- Micro processor offers a better price and performance than a main frame computers.

Disadvantage :-

1. Security :- If one system gets infected whole network will be infected.
2. Networking :- if the network get saturated than problem will transmisson.
3. Software :- Very little software to support for distributed southing system.
4. Tembal southing :- it is more difficult in trible because the analysis more required connected to more remote node or inspecting communication between node.

01-03-2020

3. Multi programming O.S :- The objective of multi programming O.S is to increase the system utilisation efficiency. The batch processing system tries to reduce the ideal lines through operator interaction. However it cannot reduce the ideal time due to I/O operations. multi programming O.S tries to eliminate ideal times by providing multiple computational task for the CPU to perform. This is achieved by keeping multiple jobs in the main stack. So, when the job is being currently executed on the C.P.U needs some I/O. So, the C.P.U passes its requirement to the IOP. Till the time I/O operation is being carried out.

The CPU is free carried some other ~~f~~ jobs.
Popular micro programming O.S are WINDOWS, LINUX, winware.
A micro programming supervisor has very difficult job of managing all the activites that take place simultaneously in the system.

4. Multithreaded OS :- When several task known as process may be run at the same time. The application consists of a sequence of instructions known as threads. These threads will be alternately active standby, suppended, or destroyed according to the priority or may be run simultaneously.

5. Multitasking & Multiprocessing O.S :-

Multitasking :- When a single program is allow to run at a time. The system is grouped under a single tasking system. While in case the O.S allows the multiple task at one time. It is classified as multitasking O.S. The resource are made to work constantly. The card reader seeds one job after the other is stored in main memory. The C.P.U switch on task to another for reading, processing and outputing. The main memory is partition and many jobs are handled simultaneously. This evabled more program to residue in the central memory at the same time instruction are execution by the C.P.U.

03-01-2020

6. Multiuser system :- RTS, parallel system.

It allows more than one user excess the system at one time. access to the computer system is normally provided via a network so that user access the computer remotly using ~~the computer~~ it terminal. In the early days of large multi user computers multiple terminals such as keyboard and associates monitor work provided. These terminals sends their command to the main multi user computer for processing and the result by displayed on the associated terminal moniter screen. Today these terminals are generally personal computers an use a network to send and receive information to the multi user computer system.
eg :- LINUX, UNIX etc.

Advantages :-

1. Cost divided among user.
2. ~~Privacy send~~
2. Better user of sources.
3. Maximum utilization of resources.

Disadvantages :-

1. Performance become slower and slower
2.
7. Time sharing O.S :- It is the logical extension of multi programming multiple jobs are executed by the C.P.U switching between them but the switching occurs so frequently that the user may interact with each program while it is running. Processing time is shared

among multiple users and interactive system provides online communication between the user and the system. The user gives instructions to the O.S or to a program directly and received and immediate response. It uses C.P.U sheduling and multi programming to provide each user with a small portion of a time shared computer. each user has atleast one sheprate program in memory. It allows many users to share computer simuntaneously.

Advantages:-

1. Easy to use.
2. User friendly.
3. No need to know any tecnical language.
4. Its the platform of all programs.

disadvantage:-

1. If any problem affected in O.S you may lose all the contents which have been stored already.
2. ~~Unknow~~
3. Unwanted user can access your system.

8. RTS:- It is used when their are a regid time requirements on the operation of a processors or the flow of data. it is often used as a control device in a dedicated application. System that controls scientific experiments medical imaging industrial control system are real time system. it is well defined fixed time contrains processing must be done within define constant. or the system will fail.

O.S components and service of an computers

eg:- Mesial longing operation.

There are two type of system.

1. ~~Hardsrove typ~~
1. Hard real time system.
2. Soft real time system.

1. Hard real time system gouranteed that critical task complete on time

2. A critical real time task gives priority over _____ or other task and retain that priority untill if completes.

Advantages:-

1. Where there are regid time requirements only RTOS gives optimal result. for small operation it
2. is most cost effective.

disvantages:-

1. It is very time specific.

04-01-2020

M.Imp

O.S components:-

External

1. ~~Process management~~ :- O.S is a very complex system to understand it we will partion it major parts.

1. Process management :- A process can be though of a program in execution. A process needs certain resources including C.P.U time memory, File and I/O devices to acomples its task. When the process terminates the O.S will reclaim any renewable resources. the process management in O.S

allocation
linking
Reallocation
load

responsible for the creation or deletion of both user and system process : it had suspention and resumption of process. It gives the provison of mechanisms for process synchronization communication and deadlock handly.

2. Main Memory Management :- It is repositly of quickly accessible data shared by the C.P.U and I/O devices. It keep track of parts of memory which are currently been used an by whom. It alocates and dealocate ~~it~~ memory space as needed. it decide which ~~dei~~ process are to reloaded into memory. when memory space become available.

3. File management :- It can store information on several different time of physical media. Such as megnetic disk tape, optical disk and megnetic disk. File is a collection of related information by user. It represent program and data. it helps in creation and deletion of files and directory. It supports the primitives for manipulating files and directory. It manages the backup of file on stable storage media.

4. I/O system management :- One of the purpose of an O.S is to hide the paculanities of specific hardware devices from the user. The I/O system consist of -
1. A memory management include buffering and caching.
2. A general device driver code.
3. Drivers for specific hardware devices.

5. Secondary storage management :- Main memory is to small to accomodate all program and data and its data are lost when power is lost. The computer system must provide secondary storage device to main memory. The management or secondary management in O.S is responsible for free space management. (storage allocation and disk allocation.)

6. Networking :- A distributed system is a collection of processors that don't share memory, peripheral devices or clock. The processors in the system are connected to a communication network which can be configured a no. of different ways. The communication network design must consider rounding and problems of security.

7. Protection system :- If a computer system has a multiple users and allows the conferent of multi processor. Then the various process must be protected from one other activites. Protection refers to a mechanism for controlling the access of program, process of user to the resource defined by a computer system.

8. Command interpreter system :- It is the interface between the user and the O.S. Some O.S include the command interpreter in the kernel. Other O.S like MS DOS, Linux treat the command interpreter as a special program i.e running when job is initiated. command interpreter refers to shells.

Scanned with CamScanner

New
Ready
Running
terminated
waiting

dinning pho

fork process (study)
Gate, competive

05/2/2020

Date / /
Page

Services of an O.S :-

1. Program development :-

(i) fisiblity
(ii) requirement Analysis.
(iii) designing
(iv) Coding
(v) Testing
(a) Valitation
(b) Verification

2. Program execution.
3. Access to I/O devices.
4. Controlled access to files.
5. System access
6. Error detection & Response
7. Accounting.

System call :-

Concurrent processes :- A fundamental function of an O.S.

Sleeping barber problem
dinning philosopher problem] Assignment - 1

12-2-2020 Fr

Concurrent processes :- Currently a computer system allow multiple programs to be loaded into memory and executed concurrently. This evolution required control of the various programs and these needs resulted in the location of a process which is a program in execution. A fundamental function of an O.S is the execution of a program and an executing program is known as process. A program is a static thing in most O.S a program is stored in

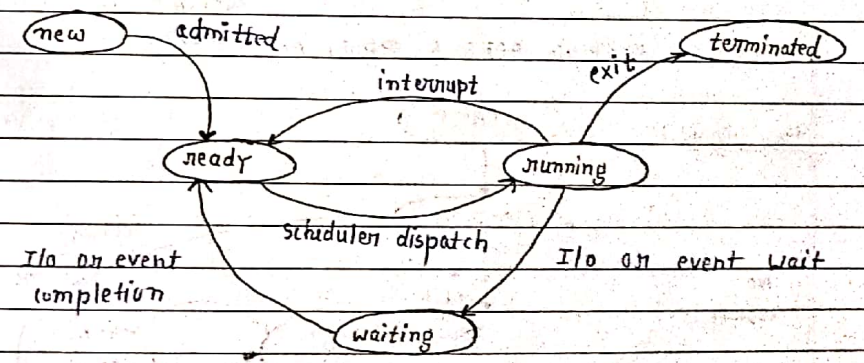
ctrl + F + delete

Date / /
Page

a file on secondary storage. Evantually an O.S is instructed to run a program. usually by a user but some times by another process. perhaps one running on other process. The O.S links that program into primary storage and begins to execute. The running entity running as process and other terms. A We will create a structure in memory where we will save the information describing the state of the process we are staping. we will call this structure a PCB (Process Control block). or (process discriptor)

Process concept :-

The Process :-



Process state :-

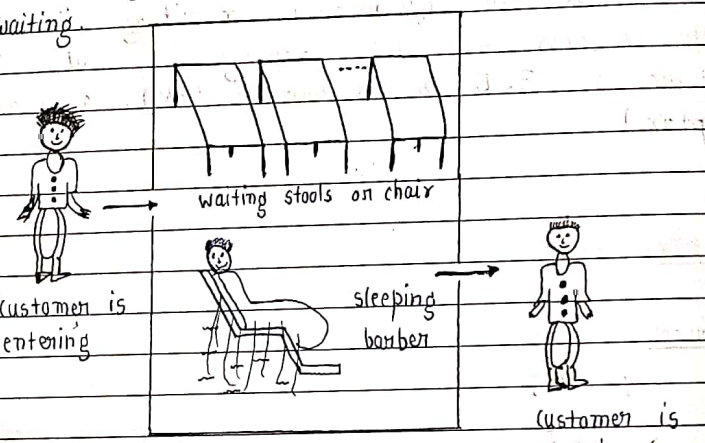
1. New :- The process is being created.
2. Running :- Instruction are being executed.
3. Waiting :- The process is waiting for some event to occur.
4. Ready :- The process is waiting to be assigned to a processor
5. Terminated :- The process has finished execution.

Scanned with CamScanner

Date / /
Page

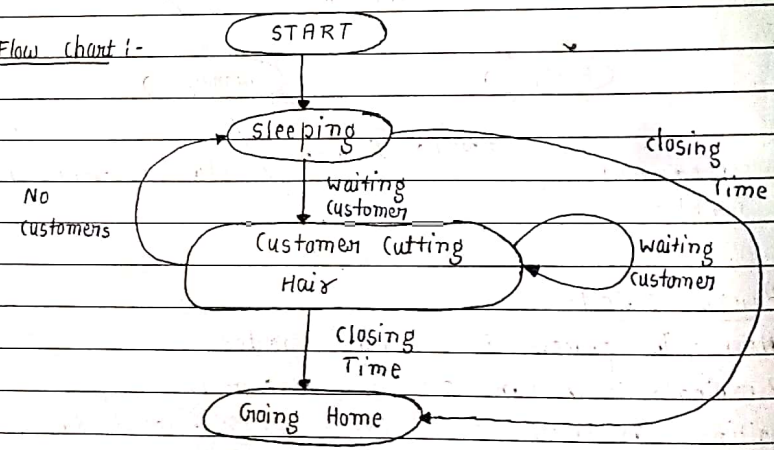
THE SLEEPING BARBER PROBLEM :- The sleeping barber problem is a classic inter-process communication and synchronization problem between multiple operating system processes.

~~Algorithm~~ :- When the barber finishes cutting a customer's hair, he dismisses the customer and then goes to the waiting room to see if there are other customers waiting.



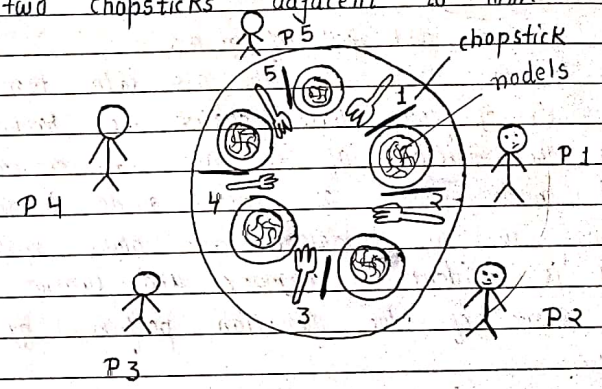
SLEEPING BARBER PROBLEM

Flow chart :-

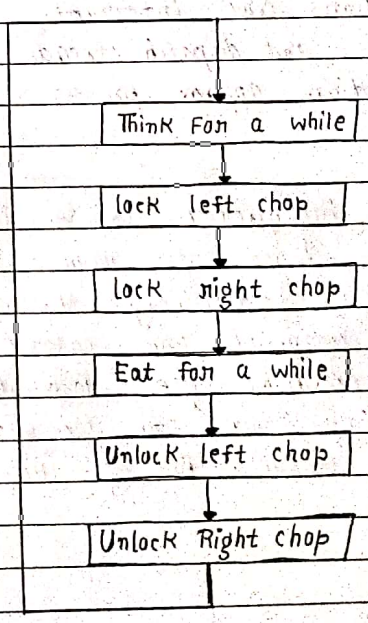


Date / /
Page

THE DINING PHILOSOPHERS PROBLEM :- The dining philosopher problem states that K philosophers seated around a circular table with one chopstick between each pair of philosophers. There is one chopstick between each ~~pair~~ philosopher. A philosopher may eat if he can pickup the two chopsticks adjacent to him.



Flow chart :-

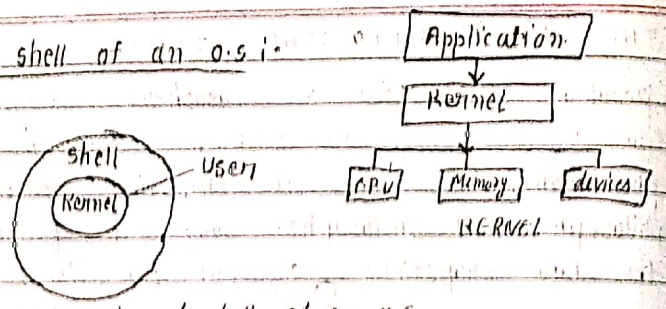


Scanned with CamScanner

13/2/2020

Imp Code

Kernel and shell of an O.S :-

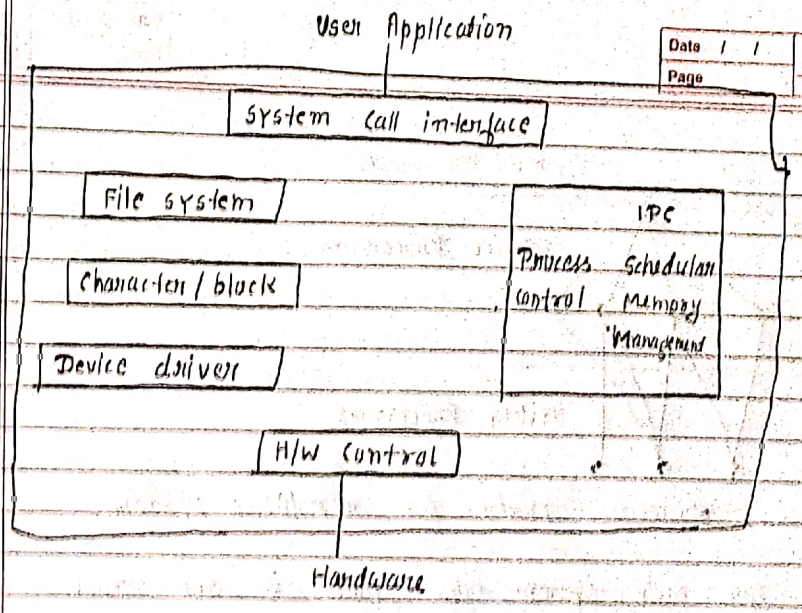


Kernel and shell of an O.S

The Kernel is the lain of the O.S code that is just about the lair of code thats shielded the hardware. Kernel is the nucleus of the O.S. It executes in prevelished mode called as Kernel mode where it has direct access to the hardware and other system software. The Kernel is resident in memory and cannot be primited. Some of the function perform by the Kernel are

1. Handle hw exceptions and interrupt.
2. Schedule priotries and dispatch treads.
3. Synchronize execution approx process in a multi processors environment.
4. The comm

The command interpreter is the interface b/w the user and the O.S hence its name Shell. The shell therefore X as an intermediaty b/w the O.S and the user using command lines entered by the letter. The shell is an executable file responsible for interpreting command transmitting them to the system and written to the result for eg: UNIX KERNEL.

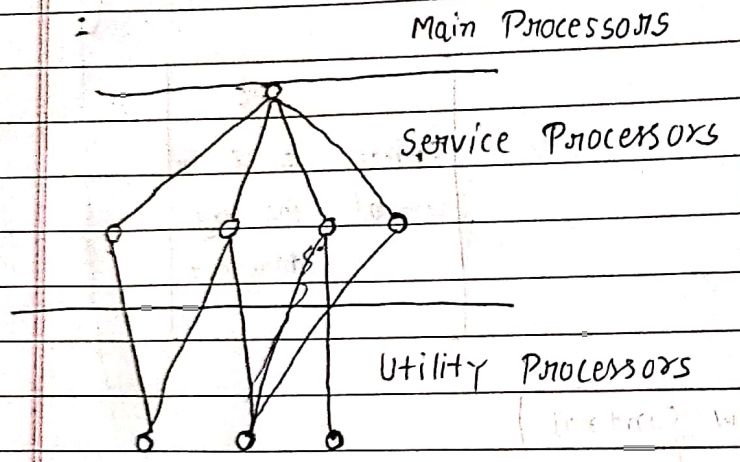


O.S Structure :-

1. Monolethic and simple structure :- The structure is that there is no structure. The O.S written as collection of procedure each of which can call any of the other once whenever it needs to this approach is well known as The big mass. System started as small symbol limited System. Even in the system it is possible to have at least a little structure. The instruction switches machine from user mode to Kernel mode and transfers control to the O.S. Basic structure of an O.S can be divided into three parts.

1. A main program that Inbox the suguested service processors.
2. A set of service processors that carry out system calls.
3. A set of utility processors that helps the service processors.

Scanned with CamScanner



structing model for monolithic system

For each system calls there is one service procedure that take care of it. The utility processors to thing that are maded by serval service processors as fetching data from user program.

eg:- MS DOS

MS DOS LAIRED STRUCTURE :-

Application program ↓
system program ↓
MS DOS device driver ↓
ROM BIUS DEVICE DRIVER ↓

MS : DOS LAYER STRUCTURE

Page fold
demand paging

Date / /
Page

constructed in this way built by EW DIJKASTRA (in 1978)
It consist C.P.U. scheduling memory management I/O buffering in the form of different lairs. lay

layered system of the O.S :- The system has 6 lairs.

layer	Function
5	The operator
4	user program
3	I/O management
2	operator process communication
1	Memory management
0	Multiprogramming and process allocation

UNIX LAYERED STRUCTURE :-

	USERS		
	SHELL & COMMANDS & INTERPRETERS	COMPILES	
	SYSTEM CALL INTERFACE TO KERNEL		
Kernel	Signal terminal handling Character I/O System Terminal Drivers	File system DISK & Tape drivers	CPU scheduling page replacement demaind paging virtual memory
	KERNEL INTERFACE TO H/W		
	Terminal controls & Terminals	device controllers disk & Tape	Memory controllers physical memory

Scanned with CamScanner

One method in the layered approach in which the O.S is broken into a no. of layers the bottom layer is the h/w. The highest layer is the user interface.

Advantages :-

1. Modular Approach :- which ~~happens~~ helps in debugging & verification of system.
2. A layer is design in such a way that it uses operation and services only of a layer below it contains them.

Disadvantages :-

1. Definition of new layer i.e How is different one level from another.
2. Since a layer can use service of a layer below it must be design carefully.

O.S Services :-

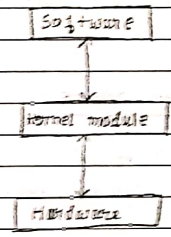
1. Program development
2. Program execution
3. Access to I/O devices
4. Controlled access to files
5. System access
6. Error detection and Response
7. Accounting

14/2/2020

Q. Difference micro / Monolithic Kernel
Define micro
Mono

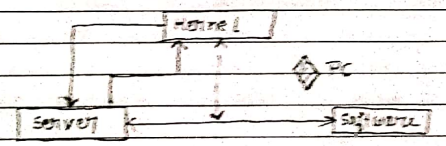
Monolithic Kernels System :- The monolithic approach defines a high level virtual interface over the h/w with a set of primitives or system calls to implement O.S devices such as process management consistency and memory management in several modules that run in supervisor mode.

Monolithic Kernel



In a monolithic kernel all O.S services run along with the main kernel thread thus also residing in the same memory area. The main advantages are kernel & dependences between system components.

Micro Kernels Or micro kernel system :- In this approach the kernel itself only provides basic functionality that allows the execution of servers. Separate programs that assume former kernel function such as device drivers and host server etc.



Micro Kernel

Scanned with CamScanner

Date / /
Page

The micro Kernel approach consist of defining a simple abstraction on the h/w with a set of primitives or system calls to implement minimal o.s services such as memory management, multi tasking and inter process communication. Micro Kernels are easier to maintain than monolethic Kernels.

One advantage of this Kernel is that of simplicity.

Differen b/w monolethic and microlethic Kernels :-

Monolethic	Microlethic
1. It is a large program which contains different modules and they can interact each other.	1. Micro Kernel contain different modules.
2. New modules can be dynamically or inserted in it at run time.	2. These modules can communicate those module which are link with it. we can only statically insert a new module in it.
3. A monolethic Kernel is one where all o.s code is in a single executable image and all o.s code runs in system mode.	3. A microKernel is a minimal o.s that perform only the essential functions of an o.s all other o.s function are performed by system processes.
4. In monolethic Kernel all o.s services runs along with the main Kernel thread thus also residing in the same memory area.	4. Micro Kernel mainly used in embaded system such as ATX o.s and symbian o.s

eg:- UNIX, MS.DOS, Solerecies.

(Automata) Adesh Pandey & Putumkar Aboulma shetty

Ten & bomb

Peter ling (Automata)

Date / /
Page

Re entrant Kernel :- A reentrant Kernel enable processes to give away the CPU while in Kernel mode not indering other processes from also entering Kernel mode. A typical used case is IO wait with a reentrant Kernel the shedular will asign the C.P.U to another process untill and interrupt from the disk controller indicate that the data is available and our thredd can be resume. if the Kernel is not reentrant a process can only be suspended while it is in user mode. All UNIX Kernel are Re entrant this means that serval processes may be executed in Kernel mode at the same time. Offcourse All UNIX processor systems only one process can progress but many can be block in Kernel mode when waiting for the C.P.U or the completion of some I/o operation. One way to provide reentrancy is to write function so that they modify only local variable and donot older global ds. Such function are called Re entrant Kernel.

15-2-2020

System calls :- Modern processors provide instruction that can be used as System calls. It provide a interface between the process and the o.s. A System call instruction is an instruction that generates an interrupt that calls the o.s. to gain control of the processor.

(Not syllabus)

Scanned with CamScanner

UNIT-2

15/2/2020

Concerent processes :-

Race condition :- A race condition occurs when multiple process or threads read and write data items so that the final result depends on the order of execution of instructions in the multiple processes.

For eg :- Two processes P_1 and P_2 share the global variable x in the execution process P_1 update variable x ~~two~~ the value two and ~~addst~~ same points its execution P_2 update x to the value three. These two task are in a race to write variable x. In this the process that updates last determine the final value of x.

For eg :- If two process P_3 and P_4 shares two global variable X and Y with initial value $x=2$, $y=3$ while in the execution process P_3 executes the assignment $X = X + Y$ and process P_4 executes the assignment $Y = X + Y$. Both the process update different variable the final value of the two variables depends on the order in which the two processes executes ~~with~~ these two assignment. If process P_3 executes its assignment statement 1st. Then the final values of $X = 5$, $Y = 8$. If process P_4 executes its assignment statement 1st. Then the final values of $x = 7$, $y = 5$.

17-2-2020

Not syllabus

System calls work in following ways :- ~~First~~ 1. the program executes the system call instruction.

2. The h/w saves the current instruction and PSW (program status word) register is in the ii and

Sudo Code

device drivers
Systemsoftware (John Donowal)

~~First~~ i PSW register.

3. 0 value is loaded into PSW register by hardware. it keeps the machine in system mode with interrupt this abled.

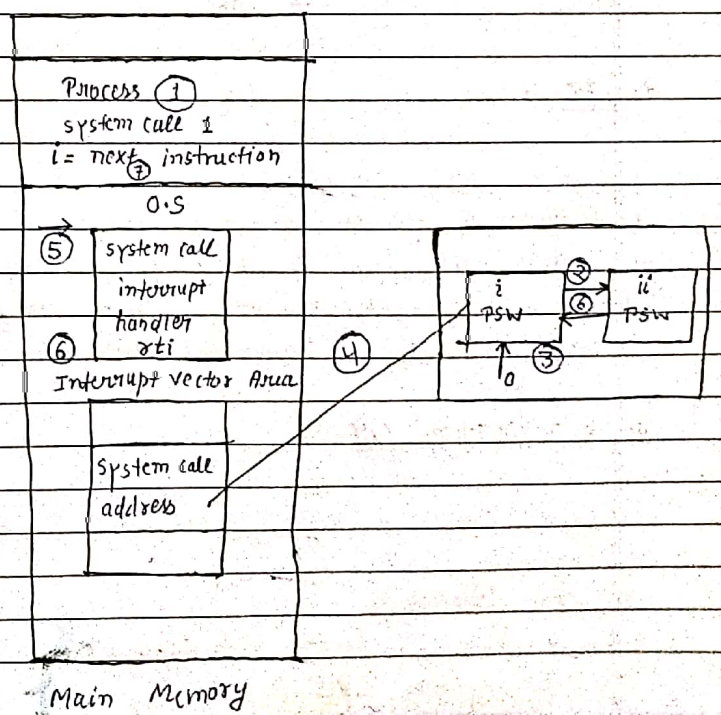
4. The h/w loads the i register from the system call interrupt vector location. This completes the execution of the system call instruction by the h/w.

5. Instruction execution ~~a~~ continues at the begining of the system call interrupt handler.

6. The system call handler completes ~~the execution~~ and executes a return from interrupt instruction. This restores the i and PSW from the ii and iPSW.

7. The ~~prog~~ process that executed the system call instruction continues at the instruction after the system call.

Flow control of system call :-



Main Memory

PSW is the bank of status word.

Type of system call :-

1. File management
2. Interprocess communication.
3. Process management.
4. I/O device Management
5. Information Maintainance

कौन से task perform

18-2-2021 File management

1. Create :- Create a new file and open it
2. delete
3. Open
4. close :- It indicates that the file is no longer using it.
5. Read / write :-
6. Stead :- Get info about a file.
7. Unline :- Remove a file from a directory.
8. Get file attribute and set file attribute.

2. IPC :-

1. Create msg queue :-
2. Create a queue to hold msg.
3. Send a msg to a msg queue. (send msg)
3. Receive msg :- Receive a msg from msg queue.
4. Close collection :- Terminates the communication

3. PM system call :-

1. Creat Process
2. Terminate Process.
3. wait Process.
4. Fork Process :- Create a duplicate of the process making a system call.

Process concept
Diagram

Quantum - Time

UNIT 1, 2 SESSIONAL-1

ACID
Atomicity

Arrival Time
Burst time
Expection Time

(Mutex = semephor = technique to solve dead lock)

5. HAND :- Hold the process.
6. ABORT :- Hold the process execution abnormally
7. LOAD :- load the process into memory.
8. EXECUTE :- execute the loaded process.
9. Get process attributes and set process attributes.
10. Allocate and ~~memo~~ free memory.

4. I/O device system call :-

1. Request device
2. Realease device
3. Read / write
4. Stead

5. TM :-

1. Get time and date
2. set time and date
3. set process time and device attributes.
4. Get process
5. Get system data and set time data.

A System calls is the description of the set of system call implemented by the o.s.

19-2-2020

Process concept :-

Requirement of mutual exclussion :-

A process have been given many definition for stance. ~~A progr~~

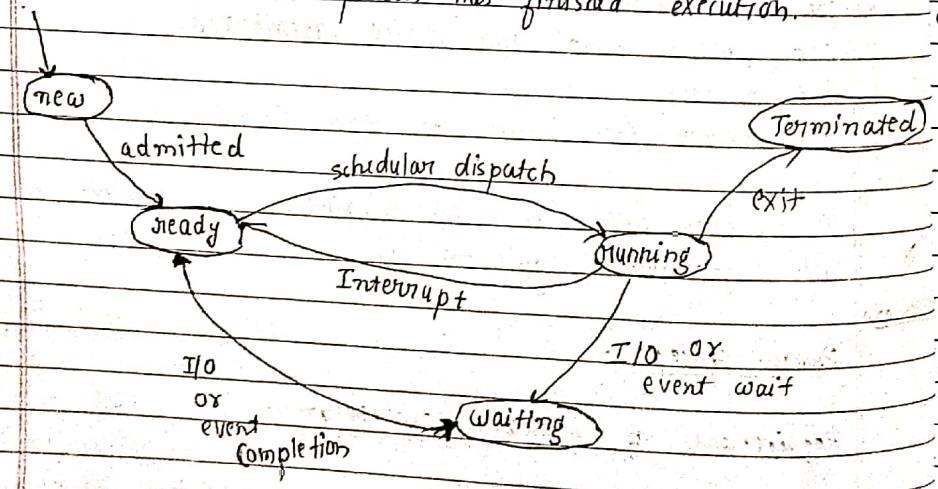
Scanned with CamScanner

1. A program in execution.
2. An asynchronous activity see the entity to which the processor are assign.
3. The dispatchable unit :- A program or unit of job or task at the time of execution is known as process. it includes all the supporting current activities. such as memory management, I/O management and C.P.U. Management. Process is a dynamic entity that is a program in execution. A process is a sequence of instruction execution.

13.

Process states :- As a process executes its change state each process may be in one of the following state.

1. New :- The process is being created.
2. Running :- The instruction are being executed.
3. Waiting :- The process is waiting for some event to occur. such as an I/O completion or reception of a signal.
4. Ready :- The process is waiting to be assigned to a processor.
5. Terminated. :- The process has finished execution.



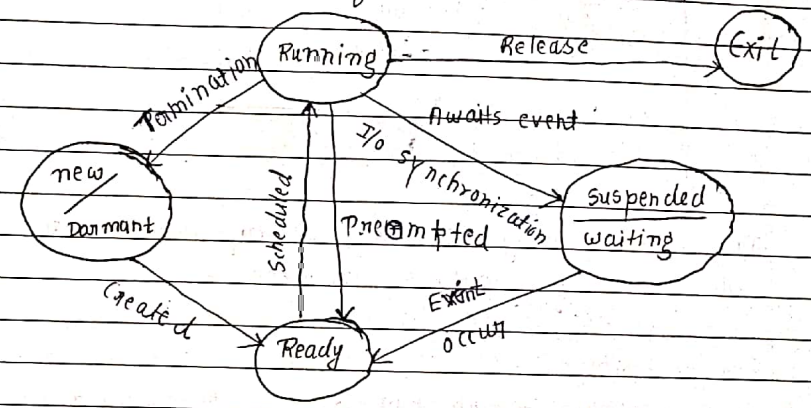
(Five state process model)

Word power made easy | RS Agarwal Reasoning | Process control

Characteristics of suspend state, Reason for process suspension, Process operation,

End semester 5-Mark

Process transition diagram :- Transition means change of one state of process to other state. A process state transition is dynamic change in the status of process. Process transition diagram also consist of 5 states.



1. New :- Process is being created.
2. Ready :- Process processes all resources accept the processor.
Running :- Process possess all resources including all processor.
Suspended :- Process lacks some resources other than processor such as synchronization symbol so that process suspended.
Waiting :- The process is waiting for some event to occur.

20-7-2020

PCB (process control block) :- Each process is represented in O.S. by a PCB also called a task control block.

Scanned with CamScanner

Pointer	Process state
Process number	
Program counter	
C.P.U. register	
Memory allocation	
event information	
list of open files	
—	
P.C.B	

It contains many pieces of information absolute with a specific process includes —

1. Process state
2. Process number :- Each process is represented by its process id.
3. Program counter :- Address of nxt instruction to be executed.
4. C.P.U. register :- The register vary in number and type depending on computer architechture they include accumulator, index register, stack pointer, general purpose register, C.P.U sheduling information :- It includes a process priority. Pointers to sheduling queues and any other sheduling parameters.
5. Memory allocation :- It includes the information about the memory capacity depending on the memory status used by the o.s.
6. Accounting information :- The information includes the amount of C.P.U and real time use.

I/o status information :- It includes the list of I/o devices allow allocated to the process a

Principle of Concurrency, Mutual exclusion problem critical section problem, PISEDUO code process synchronization concurrency.

list of open files and so on.

Concurrent process :- The coprnation, communication and competation between processes that either run simantanously or a interlived in arbitary ways to give the appearance of running symantanously and optimize the working of the computer this is the concept concurrency. These process executing concurrently are either —

1. Competing process.
2. Co-oprating process.

1. Competing process :- Process that do not exchange information (independent) can not affect the execution of each other but they can compete for devices and other resources. The o.s. carefully allocate resources and properly visolate processes from each other.

2. Co-operating (dependent) processes :- Those processes which work together in one of the following two ways —

a. directly :- By exchange messages.
b. indirectly :- By sharing common memory. process can cooperate each other.

Principle of concurrency :- Concurrency is the interleaving of process in time to give the appearance of simuntaneous execution.

eg :-
```
chin = getchar ();
chout = chin ;
put char = (chout) ;
```

Scanned with CamScanner

if two process P_1 and P_2 both executing this code at the same time.

1. P_1 enters this code but interrupted after reading the character x into chin.
2. P_2 enters this code and runs it to completion reading and displaying the character Y.
3. P_1 is resume but chin now contains the character Y so P_1 displays the wrong character the general solⁿ is to allow only one process at a time to enter the code that access chin such code is often called a critical section. (When one process is inside the critical section of code other process must be prevented from entering this section this requirement is known as mutual exclusion.)

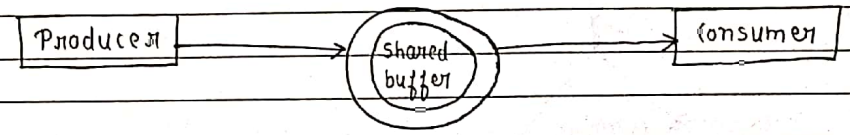
Critical section problem :- A critical section is a sequence or block of code run by a thread process that references one or more variables in a read write or update fashion while any of those variable possibly being altered or changed by another thread or process. Requirement for a valid solⁿ to the critical section problem.

1. Mutual exclusion :- At any time at most one process can be in its critical section. Mutual exclusion is in many ways the fundamental issue in concurrency. it is the requirement that when a process 'p' is accessing the shared resources R no other process should be able to access R until p has finished with R.
   eg:- Printer, I/O devices, shared data structure.

Critical section :- It is a sequence of block or code run by a thread or process that references one or more variables.

Imp Producer consumer problem :- It is a classical problem of synchronization and it is the problem of critical section. A producer who produces the item in the buffer and consumer who consume the item from the buffer that is why it is known as bounded buffer problem for this type of senario synchronization is required we must satisfied the two condition -

1. The consumer must wait if the buffer is empty.
2. The producer must wait if the buffer is full.



26-2-2020 Pre sudo code (Producer consumer problem) :-

Producer :-

```
Producer :
While (true)
{
/* Produce an item and put in next produced */
    While (count == BUFFER SIZE) : // do nothing
    buffer [in] = next produced ;
    in = (in + 1) % BUFFER - SIZE ;
    count ++ ;
}

Consumer :
    While (true)
{
While (count == 0) // do nothing
next consumed = buffer [out] ;
out = (out + 1) % BUFFER - SIZE ;
```

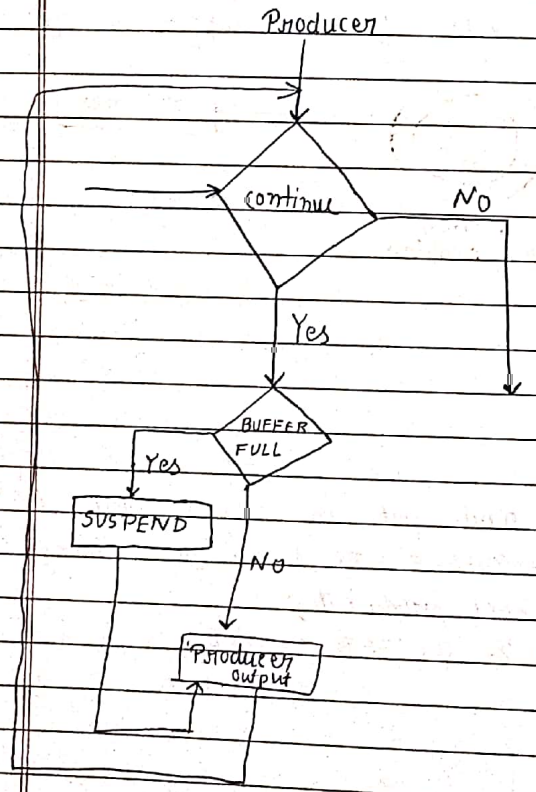
Mannat Wannut
(In phone use)

Date / /
Page

COUNT --;
// consumer the item in the next consumed.
}

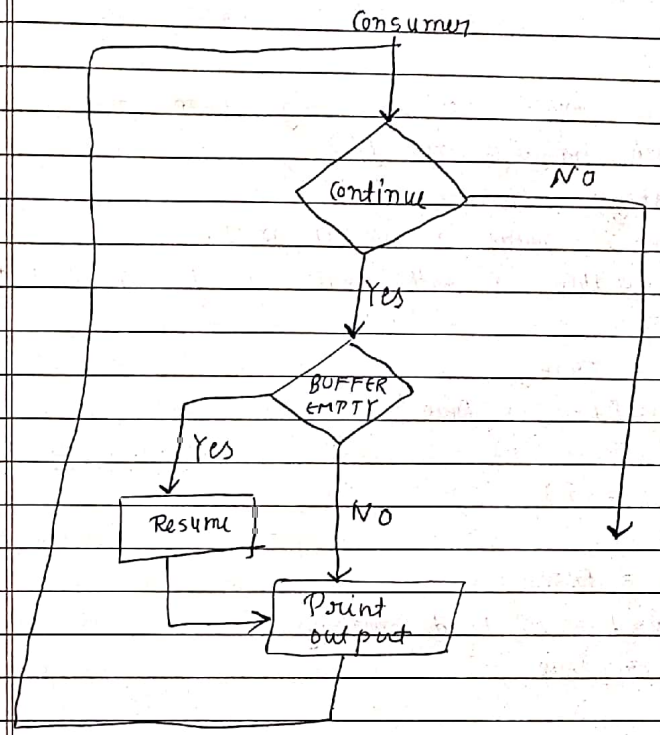
Vector solution :-

Flow chart :-



Explain the software soln for critical section Vectors, Peterson

Date / /
Page



Deker or peterson soln :-

This is the first collect soln propost the two process case. The requesting process set our FLAG and then check our neighbours flag. if that flag is also set the turn variable is used. There is ~~too~~ no progress problem now because we no the other process is in or before its critical reason. if the turn is ours. we wait for the FLAG of the other process to clear. No process will wait indefinitely with its flag set and it is a software soln of critical section or producer consumer or mutual exclusion problem.

Scanned with CamScanner

Date / /
Page

Algorithm :-

Global boolean array variable flag with two components both intialize to false ;
Global integer turn = 0 ;
process P ; where i is 0 or 1.
integer variable i with value 0 for P_0 and 1 for P_1 ;

```
flag [i] = True ;
While (flags [1-i] = True)
{
  if (Turn ! = i)
  }
} Flags [i] = false ;
While (Turn ! = i) // do nothing
flags [i] = True ; }
}
// Critical section
Turn = i-1 ;
flags [i] = False ;
// Remainder region
} While (True) ;
```

Semaphores :- Hardware based solⁿ to the critical section problem. it is called the synchronization tool. A semaphore s is an int. variable that apart from intialization is accessed only through to standard atomic operations.

1. Wait () operation :- Termed as P ()
if the semaphore value is big enough reduced it by one. A process or thread does a wait before accessing a shared resource or to wait

Peterson Problem

Sema phor

Date / /
Page

for any other signal from another process or thread.

Definition of wait () :-

```
wait (s) {
        while s <= 0
        ; // no-operation
        s - - ;
    }
```

2. Signal operation :- Termed as V ()

Increase the semaphore value by one. A process or thread does a signal after accessing a shared resource or to send any other signal to another process or thread.

Defination of signal s() :-

```
signal (s)
{
  s++ ;
}
```

semaphore variant :- There are two types of semaphore variant.

1. counting semaphores variant :- The value of a counting semaphores can range over an unrestricted domain counting semaphores can be used to control access to a given resource consisting of a finite number of stances. The semaphore is intialize to the no. of resources available.

Scanned with CamScanner

2. Binary semaphores :- The value of this can range only b/w 0 and 1. on some system binary semaphores are known as mutex locks. As they are locks that provide mutual exclusion. we can use the binary semaphores to deal with the critical section problem for multiple processes.

Mutual exclusion implementation with semaphore :-

```
do
{
wait (mutex) :
// critical such as
    signal (mutex);
// remainder section
}
while (true);
```

27-2-2020 Test and set and swap instructions :- Software based soln such as Peterson are not guaranted to work on modern computer architecture. Instead we can generally states that any soln to the critical section problem requires a simple lock. Race conditions are prevented by requiring that critical region be protected by locks.
The important characterstics is that this instruction atomically. Thus if two set and test instructions executes simuntaneously each on a different CPU. They will be executed sequencely in some arbitary order. If the machine supports the test and set instruction then we can impliment mutual exclusion by declaring a boolean variable lock. initialized to false.

```
do
{
While (Test and set lock (& lock))
    ; do nothing
lock = false;
// remainder section.
} while (True);
```

The swap instruction in contrast the test and set instruction operates on the contents of two words. like the test and set instruction. It is executed atomically.

```
While swap (boolean *a, boolean *b) {
    boolean temp = *a;
        *a = *b;
        *b = temp;
}
```

Now if the machine supports the swap instructions then the mutual exclusion can be provided as a global boolean variable lock is declared and initialize to false.

Mutual exclusion implementation with the swap instructions:-

```
do
{
Key = TRUE;
While (Key == TRUE)
swap (& lock, & Key);
    // critical section
```

Scanned with CamScanner

```
lock = FALSE;
// remainder section
}
while (TRUE);
```

Scanned with CamScanner