ASSIGNMENT

OF

OPERATING SYSTEM

PREPAIRED BY:-

ARIHANT JAIN

B.TECH (CSA) Vth SEM

ROLL NO. 38

ANSWER 1:-

The five main functions of an OS are:

* **Process Management**: It manages all the process from start to shut down of a computer. It handles creation and deletion of user and system processes.
* **Memory Management:** It decide which process are loaded in the memory when memory space becomes available. It allocate and deallocate the memory space as needed.
* **File Management:** It create files and directories ,rename ,move, copy and delete files.
* **Security Management:** It maintain the security of each processes by providing a mutual exclusion.
* **Command interpreter:** A command interpreter is an interface between system and the user. There are two types of interface:

i)Command Line Interface

ii)Graphic User Interface

ANSWER 2:-

Some programs are designed to make use of CPU idle time, meaning that they run at a low priority so as not to impact programs that run at normal priority. This allows non-crucial background programs to only run when it would not affect the performance of other applications. Programs like this may cause the CPU to be at 100% utilization all the time which causes the CPU to consume more power, since most modern CPUs can enter power-save modes when they are completely idle. Modern processors use idle time to save power. Common methods are reducing the clock speed along with the CPU voltage and sending parts of the processor into a sleep state.

ANSWER 3:-

int main() // CPU execution

{

int i, int j; //CPU execution

scanf(“%d”, &i); // I/O execution

for(j=0; j<i; j++)

CPU execution

{

sum= j+i;

}

printf(“%d”, sum); // I/O execution

exit(0); } // CPU execution

ANSWER 4:-

Multitasking=Multitasking is the ability of an operating system to execute more than one task simultaneously on a single processor machine. Though we say so but in reality no two tasks on a single processor machine can be executed at the same time. Actually CPU switches from one task to the next task so quickly that appears as if all the tasks are executing at the same time. More than one task/program/job/process can reside into the same CPU at one point of time.

Multiprocesing=Multiprocessing is the ability of an operating system to execute more than one process simultaneously on a multi processor machine. In this, a computer uses more than one CPU at a time.

Multiprogramming=Multiprogramming is also the ability of an operating system to execute more than one program on a single processor machine. More than one task/program/job/process can reside into the main memory at one point of time. A computer running excel and firefox browser simultaneously is an example of multiprogramming.

ANSWER 5:-

A program is a set of instructions that are to perform a designated task, where as the process is an operation which takes the given instructions and perform the manipulations as per the code, called ‘execution of instructions’. A process is entirely dependent of a ‘program’.  
  
A process is a module that executes modules concurrently. They are separate loadable modules. Where as the program perform the tasks directly relating to an operation of a user like word processing, executing presentation software etc.

ANSWER 6:-

**NEW**: secondary memory

**READY**: main memory

**RUN**: main memory

**BLOCK**: main memory

**TERMINATE**: secondary memory

**SUSPEND READY**: secondary memory

**SUSPEND WAIT** : secondary memory

ANSWER 7:-

Context switching is the procedure of storing the state of an active process for the CPU when it has to start executing a new one. For example, process A with its address space and stack is currently being executed by the CPU and there is a system call to jump to a higher priority process B; the CPU needs to remember the current state of the process A so that it can suspend its operation, begin executing the new process B and when done, return to its previously executing process A.

ANSWER 8:-

|  |  |  |  |
| --- | --- | --- | --- |
| Minimum | | Maximum | |
| Ready | 0 | | M |
| Running | 0 | | N |
| Block | 0 | | M |

n=CPU,m=process

ANSWER 9:-

**Long-term scheduler (or job scheduler)** -

–selects which processes should be brought into the ready queue.

–invoked very infrequently (seconds, minutes); may be slow.

–controls the degree of multiprogramming

**Short term scheduler (or CPU scheduler) -**

–selects which process should execute next and allocates CPU.

–invoked very frequently (milliseconds) - must be very fast

**Medium Term Scheduler**

–swaps out process temporarily

–balances load for better throughput

ANSWER 10:-

**1. Match User Experience and Expectations**By matching the sequence of steps, layout of information and terminology used with the expectations and prior experiences of the user, the friction and discomfort of learning a new system will be reduced.

**2. Consistency**As well as **matching people’s expectations** through terminology, layout and interactions the way in which they are used should be consistent throughout the process and between related applications.

**3. Functional Minimalism**

The range of possible actions should be no more than is absolutely necessary. Providing too many options can detract from the primary functions and reduce usability by overwhelming the user with choices.

* Avoid unnecessary features and functions
* Break complex tasks into manageable sub-tasks
* Limit functions rather than the user experience.

**4. Control, Trust and Explorability**These three elements are fundamentally important to any system. If users feel in **control of the process** they will be more comfortable using the system. If the user is comfortable and in control they will **trust that the system will protect them** from making unrecoverable or unrecognized errors or from feeling stupid. Trust inspires confidence and **with confidence the user is free to explore** further.

**5. Error Prevention, Detection and Recovery.**The best way to reduce the amount of errors a user makes is to anticipate possible mistakes and prevent them from happening in the first place. If the errors are unavoidable we need to make them easy to spot and help the user to recover from them quickly and without unnecessary friction.

**Error Prevention**  
Prevent errors by:

* Disabling functions that aren’t relevant to the user
* Using appropriate controls to constrain inputs (e.g. radio buttons, dropdowns)
* Providing descriptive, clear instructions and considering preemptive help
* As a last resort provide clear warning messages

**Error Detection**  
Anticipate possible errors and provide feedback that helps users verify that:

* They’ve done what they intended to do
* What they intended to do was correct

It is important to remember that providing feedback by changing the visual state of an object or item is more noticeable than a written message.

**Error Recovery**  
If the error is unavoidable provide clearly marked ways for the user to recover from it. For example provide “back”, “undo” or “cancel” commands.

ANSWER 11:-

A virus cant just take hold of the system and keep running as since many process run through the CPU according to their priority following different scheduling algorithms and a virus cant just suspend all of the process to run itself. No process other than a system process gets this much priority and thus a virus cant just take over a system and keep running to freeze the other processes. So the Virus program first needs to get into the Motherboard firmware (BIOS) to take control of the whole system. But currently there are no malwares known to the world which could do such complicated things. It doesn’t mean that they do not exist. A CPU is at the lowest end of all computer components. It is coded in a different language then your operating system (CPU is coded in Assembly and OS mainly in C++, the virus has to take over loads of parts before it can reach the CPU.

ANSWER 12:-

Dispatcher is a module that gives control of the CPU to the process selected by the Short Term Scheduler. It receives control in kernel mode as the result of an interrupt or system call.

ANSWER 13:-

Applications of Real Time operating system:

* Almost all the modern telecommunication systems make use of RTOS.
* Radar systems ,network switching control systems, satellite monitoring systems, satellite launch-control and maneuvering mechanisms , global positioning systems all have their roots in RTOS.
* Now a days RTOs are increasingly finding use in strategic and military operations. These are used in guided missile launching units, track-and-trace spy satellites, etc.

ANSWER 14:-

A system call, sometimes referred as a kernel call, is a request made via a software interrupt by an active process for a service performed by a kernel.

ANSWER 15:-

**fork() system call:**

When fork() is called ,the OS will create a new process that is exactly the same as the parent process .This means all the state that was there previously is copied including open files, register state and all memory allocations which include the program code. The return value from the system call decide whether the new process is created or not. The return value from the parent process will the process ID of the child.

**exec() system call:**

Through fork() the existing process can create a new process but with exec() the new process is not part of the same program as parent process. This is the case in the shell, when a user starts a command it needs to run in a new process. exec() will replace the contents of the currently running process with the information from a program binary.