

CS 2028 -UNIX INTERNAL

UNIT-1**1. Define shell?**

A shell is a program that provides the traditional, text-only user interface for Linux and other Unix-like operating systems. Its primary function is to read commands that are typed into a console.

2. Mention the properties of UNIX OS?

- It has a simple user interface that has the power to provide the services that users want.
- It provides primitives that permit complex programs to be built from simpler programs.
- It uses a hierarchical file system that allows easy maintenance and efficient implementation.
- It uses a consistent format for files, the byte stream, making application programs easier to write.
- It provides a simple, consistent interface to peripheral devices.
- It is a multi-user, multiprocess system; each user can execute several processes simultaneously.
- It hides the machine architecture from the user, making it easier to write programs that run on different hardware implementations.

3. Mention the use of the fork System call?

The 'fork()' call is used to create a new process from an existing process. The new process is called the child process, and the existing process is called the parent. We can tell which is which by checking the return value from 'fork()'

4. Mention the use of command “Passwd”?

The passwd command sets and changes passwords for users. Use this command to change your own password .

5. Define Exceptions?

An exception is a runtime error or warning condition, which can be predefined or user-defined. Predefined exceptions are raised implicitly (automatically) by the runtime system.

6. Define interrupt?

Interrupt is a signal sent or given to the processor in order to interrupt or to stop the current process/execution and transfer the control to the specified process in order to perform particular task.

7. List the building block primitives of UNIX with Examples?

Unix enables to write small modular program that can be used as building block to build more complex programs. There two primitives:

I/O redirection

Pipe()

Eg: cat1 test1 > test2

8. Define Swapping?

Swapping is a useful technique that enables a computer to execute programs and manipulate data files larger than main memory.

9. Define paging?

paging is one of the memory-management schemes by which a computer can store and retrieve data from secondary storage for use in main memory. In the paging memory-management scheme, the operating system retrieves data from secondary storage in same-size blocks called pages. The main advantage of paging over memory segmentation is that it allows the physical address space of a process to be noncontiguous.

10. Characterization of unix file system?

- Hierarchical structure
- Consistent treatment of file data.
- Ability to create and delete files.
- Dynamic growth of files.

11. Define user mode?

It is a non-privileged mode in which each process starts out. It is non-privileged in that it is forbidden for processes in this mode to access those portions of memory that have been allocated to the kernel or to other programs.

12. Define kernel mode?

When the CPU is in kernel mode, it is assumed to be executing trusted software, and thus it can execute any instructions and reference any memory addresses.

13. Define buffer?

A buffer is a region of a physical memory storage used to temporarily hold data while it is being moved from one place to another.

14. Define system call?

A system call is the mechanism used by an application program to request service from the operating system. Operating systems contain sets of routines for performing various low-level operations.

15. What are the important fields in u area?

- Pointer to the process table
- Parameters of the current system call
- File descriptors for all open file
- Internal I/O parameters

16. Define various lifetime of a process?

- The process is currently executing in user mode.
- The process is currently executing in kernel mode.
- The process is not executing, but it is ready to run as soon as the scheduler chooses it.
- The process is sleeping.

UNIT – II

1. Define pipes?

A pipe is a technique for passing information from one program process to another. Unlike other forms of interprocess communication (IPC), a pipe is one-way communication.

2. Define major & minor number?

Major and minor numbers are associated with the device special files in the /dev directory and are used by the operating system to determine the actual driver and device to be accessed by the user-level request for the special device file.

3. Define buffer catch?

When a process wants to access data from a file, the kernel brings the data into main memory, alters it and then request to save in the file system.

4. Mention the use of LSEEK System call?

lseek is a system call that is used to change the location of the read/write pointer of a file descriptor.

5. Define Inode Cache?

Inode cache cache stores logical metadata information about files (size, access time, etc). It is a linked list that stores the inodes and pointers to all pages that are part of that file and are currently in memory.

6. Define inode?

An inode is a data structure on a file system in UNIX that stores all the information about a file except its name and its actual data.

7. Define various fields in disk inode?

- File owner identifier.
- File type.
- File access permissions.
- File access time.
- File size.

8. Write the formula to compute the byte offset?

Byte offset = ((inode number-1) modulo (number of inodes per block))* size of disk inode.

9. Write the formula to compute the block number?

Block number = ((inode number-1)/number of inodes per block)+start block of inode list.

10. What are the fields in super block?

- Size of file system
- The number of free blocks in file system.
- A list of free blocks.
- The index of next free block.
- The size of inode list.
- The number of free inodes.

11. Write the syntax of open system call?

Fd = open(pathname, flags, modes);

Pathname is a file name, flags indicates the type of open, modes gives the permissions.

12. Define directories?

A directory is a file whose data is a sequence of entries, each consisting of an inode number

and the name of a file contained in the directory.

13. What is the use of iput algorithm?

The iput algorithm is used to releasing the inode. A process is waiting for the inode to become unlocked

14. What is the status of the in-core inode?

- Indicates weather
- The inode is locked
- A process is waiting for the inode to become unlocked
- The in-core representation of the inode.
- The file mounts point.

15. What is the output of bmap algorithm?

- Block number in file system.
- Byte offset into block.
- Bytes of I/O in block
- Read ahead block number.

16. What is the use of namei algorithm?

The namei algorithm is used to convert path name to inode.

17. Define super block?

A superblock is a record of the characteristics of a filesystem, including its size, the block size, the empty and the filled blocks and their respective counts, the size and location of the inode tables, the disk block map and usage information, and the size of the block groups.

18. What are the inputs of bmap algorithm?

- Inode
- Byte offset

19. What are the algorithms present in buffer allocation process?

- Getblk
- Brelse
- Bread
- Breada
- Bwrite

20. What's the use of file owner identifier?

The file owner identifier is used to identify the owner type. Inode cache cache stores logical metadata information about files.

UNIT-III

1. Write the syntax of read system call?

Number = read(fd, buffer, count)

Where fd is file descriptor, buffer is the address of data structure, count is the number of bytes that the user wants to read.

2. What are the inputs of read algorithm?

- User file descriptor
- Address of buffer in user process
- Number of bytes to read

3. What are the I/O parameters saved in U area?

- Mode
- Count
- Offset
- Address
- Flag

4. Write the syntax for write system call?

Number = write(fd, buffer, count)

Where fd is file descriptor, buffer is the address of data structure, count is the number of bytes that the user wants to read.

5. Write the syntax for lseek algorithm?

Position = lseek(fd, offset, reference);

6. Write the syntax for file creation?

Fd = create(pathname, modes);

7. What are the inputs for making the new node?

- Node
- File type
- Permissions
- Major, minor device number

8. Write the syntax for change owner and change mode?

- ❑ Chown(pathname, owner, group);
- ❑ Chmod(pathname, mode);

9. What are the types of pipe?

- ❑ Named pipe
- ❑ Unnamed pipe

10. Write the syntax for mounting a file system?

Mount(special pathname, directory pathname, options);

Where special pathname is the name of device, directory path name is the directory in the existing hierarchy.

11. What are the fields in mount table entry?

- ❑ A device number
- ❑ A pointer to a buffer
- ❑ A pointer to the inode
- ❑ A pointer to root inode

12. What are the inputs of mount algorithm?

- ❑ File name of block special file
- ❑ Directory name of mount point
- ❑ Options(read only)

13. Write syntax for pipe system call?

Pipe(fdptr)

Where fdptr is the pointer to an integer array.

14. What's the use of dup system call?

The dup system call copies a file descriptor in to the first slot of the user file descriptor table.

15. Write the syntax for dup system call?

newfd = dup(fd) CS605

16. What is the use of flag register?

The flag register indicates if address is in user or kernel memory.

17. What are the flags present in open system call?

- O_CREAT
- O_TRUNC

18. What the use of algorithm chdir?

The algorithm chdir changes the current directory of a process. The syntax is Chdir(pathname);

19. Write the syntax for STAT and FSTAT?

- Stat(pathname, statbuffer);
- Fstat(fd, statbuffer);
-

20. What are the inputs of link algorithm?

- Existing file name
- New file name

UNIT – IV

1. Define signal?

A signal is an asynchronous event which is delivered to a process.

2. Define context of the process?

A context switch (also sometimes referred to as a process switch or a task switch) is the switching of the CPU (central processing unit) from one process or thread to another.

3. Define system boot?

When the computer is switched on, it activates the memory-resident code which resides on the CPU board. The normal facilities of the operating system are not available at this stage and the computer must 'pull itself up by its own boot-straps' so to speak. This procedure therefore is often referred to as bootstrapping, also known as cold boot.

4. Define init process?

The init process is otherwise known as zombie process. In which the process will be in the main memory even the exit() system call occur.

5. What are the fields of process table?

- The process state
- Several user identifier
- Process identifier

6. Define context switch?

A context switch (also sometimes referred to as a process switch or a task switch) is the switching of the CPU (central processing unit) from one process or thread to another.

7. What is page fault?

An invalid page fault or page fault error occurs when the operating system cannot find the data in virtual memory. This usually happens when the virtual memory area, or the table that maps virtual addresses to real addresses, becomes corrupt.

8 What are the various process state?

- Process is executing in user mode.
- Process is executing in kernel mode
- The process is not executing but is ready to run as soon the kernel schedules it.
- The process is sleeping and resides in main memory.
- The process is ready to run, but the swapper must swapped the process in to main memory before kernel schedules it for execution.

9. What are the logical sections present in unix?

UNIX system consists of 3 logical sections.

- Data
- Text
- Stack

10. Define region?

A region is a contiguous area of the virtual address space of a process that can be treated as a

distinct object to be shared or protected. Eg: text, data, stack.

11. What are the components present in register context?

- The program counter- specifies the address of next instruction.
- The processor status register- specifies the H/W status of the register.
- Stack pointer- current address of the next entry in the kernel.
- The general purpose registers – contain data generated by the process during its execution.

12. What are the components present in system- level context?

- The process table entry- define the state of a process.
- Preion entries, region tables and page tables – define mapping from virtual area.
- The kernel stack – contain the stack frames of kernel procedures.

13. Under what circumstances the kernel provide context switch?

- When a process puts itself to sleep.
- When it exits

- When it returns from a system call user mode but not a eligible pro
- When it returns to user mode after completes handling an interrupt but it is not eligible process to run.

14. What are the steps for a context switch?

- Decide whether to do a context switch, and permissible now.
- Save the context of old process
- Find the best process to schedule.
- Restore its context.

15. What are the entries present in region table?

- A pointer to the inode of the file.
- The region type (text, shared memory, private data).
- The size of region.
- The location of the region in physical memory
- The status of register
- The reference count.

16. What are the operation that manipulate region ?

- Lock a region
- Unlock a region
- Allocate a region
- Attach a region to the memory space of process
- Free a region
- Load a region from a file into memory space of process.
- Detect a region from the memory space of a process

17. Write the operations that perform using fork system call?

- Allocates a slot in the process table for the new process.
- Assigns a unique ID number to a child process.
- It makes a logical copy of the context of the parent process.
- Increments file and inode table .

18. What are the input for exec system call?

- File name
- Parameter list
- Environment variable list.

19. Write the syntax for signal system call?

Oldfunction = signal(signum, function)

Where signum is the signal number the process is specifying the action, function is the address of the user function.

20. What is the use of signal and wait system call?

Wait system call allows a parent to synchronize its execution with the exit of a child process.

Signal system call informs processes of synchronous events.

UNIT – V

1. Define swapping?

Unix systems transferred entire processes between primary memory and the swap device, but did not transfer parts of a process independently, except for shared text. Such a memory management policy is called swapping.

2. What are the types of devices in unix system?

- Block devices
- Raw or character devices.

3. What are the 3 stages at which device configuration can be specified?

- i. Administrators can hard-core configuration data into files that are compiled and linked when building the kernel.
- ii. Administrator can supply configuration information after the system is already running.
- iii. Self-identifying devices permit the kernel to recognize which device is installed.

4. What are the operations specifies for device close?

- a. Search file table and make sure that no other processes still have the device open.
- b. For character device, the kernel invokes the device close procedure and return to user mode.
- c. For block device the kernel search the mount table to make sure that the device does not contain a mount file system

5. What are the functions of a line discipline?

- To parse input string in to lines
- To process erase characters
- To process a kill character that invalidates all characters typed so far on the current line.
- To echo (write)received characters to the terminal
- To expand output such as tab characters
- To generate signals to processes.

6. What are the data structures for terminal drivers associated with clists?

- i. A clist to store data for output to the terminal.
- ii. A clist to store “raw” input data provided by the terminal interrupt handler as the user type it in.
- iii. A clist to store “cooked” input data.

7. Define scheduling?

When more than one process is runnable, the operating system must decide which one first. The part of the operating system concerned with this decision is called the scheduler, and algorithm it uses is called the scheduling algorithm.

8. Define demand paging?

Demand paging is an application of virtual memory. In a system that uses demand paging, the operating system copies a diskpage into physical memory only if an attempt is made to access it.

9. Explain real time processing?

In a real time processing, there is a continual input, process and output of data. Data has to be processed in a small stipulated time period (real time), otherwise it will create problems for the system.

example: assembly line robots and radar system.

10. What's the use of clock handler?

The clock handler runs with the processor execution level set high, preventing other events from happening while handler is active.

11. What are the functions of the clock interrupt handler?

- The functions of the clock interrupt handler are
- Restart the clock
- Gather system and process accounting statistics
- Keep track of time.
- Control process scheduling
- Sent signals to processes on request.

12. What all the bit fields that are used to support demand paging?

- Valid
- Reference
- Modify
- Copy on write
- Age

13. What are the types of page fault?

- i. Validity fault
- ii. Protection fault

14. Write the states for the cause of page fault?

- i. Page is on a swap device and not in memory
- ii. Page is on the free page list in memory
- iii. Page in an executable file.
- iv. Marked “demand zero”
- v. Marked “demand fill”

15. What are the types of devices?

- Block device
- Character device

16. Write the system call for terminal polling?

Select(nfds, rfd, wfds, efds, timeout)

Where,

nfds gives the number of file descriptor

rfd, wfds, efds points to bit mask

17. Write the syntax for ptrace system call?

ptrace(cmd, pid, addr, data)

Where

cmd specifies various commands, pid is the process ID, addr is the virtual address,

data is an integer value to be written.

18. Write the syntax for msgsnd system call?

`msgsnd(msgqid, msg, count, flag);` where,

msgqid is the descriptor, count gives the size of data array, flag specifies the action.

19. What are the inputs for msgsnd algorithm?

- Message queue descriptor
- Address of message structure
- Size of message
- Flags

20. Write the syntax for shmget system call?

`Shmid = shmget(key, size, flag);`

Where, size is number of bytes in the region.

21. What is a Map?

A Map is an Array, which contains the addresses of the free space in the swap device that are allocatable resources, and the number of the resource units available there. This allows First-Fit allocation of contiguous blocks of a resource. Initially the Map contains one entry – address (block offset from the starting of the swap area) and the total number of resources. Kernel treats each unit of Map as a group of disk blocks. On the allocation and freeing of the resources Kernel updates the Map for accurate information.

22. What is a Region?

A Region is a continuous area of a process's address space (such as text, data and stack). The kernel in a 'Region Table' that is local to the process maintains region. Regions are sharable among the process.

23.Explain kernal and shell.

Kernal: It carries out basic operating system functions such as allocating memory, accessing files and handling communications. Shell:A shell provides the user interface to the kernal. There are 3 major shells : C-shell, Bourne shell , Korn shell

24.What is a Daemon?

A daemon is a process that detaches itself from the terminal and runs, disconnected, in the background, waiting for requests and responding to them. It can also be defined as the background process that does not belong to a terminal session. Many system functions are commonly performed by daemons, including the sendmail daemon, which handles mail, and the NNTP daemon, which handles USENET news. Many other daemons may exist. Some of the most common daemons are: init: Takes over the basic running of the system when the kernel has finished the boot process. inetd: Responsible for starting network services that do not have their own stand-alone daemons. For example, inetd usually takes care of incoming rlogin, telnet, and ftp connections. cron: Responsible for running repetitive tasks on a regular schedule.

25.What is a zombie?

When a program forks and the child finishes before the parent, the kernel still keeps some of its information about the child in case the parent might need it - for example, the parent may need to check the child's exit status. To be able to get this information, the parent calls `wait()'; In the interval between the child terminating and the parent calling `wait()', the child is said to be a `zombie' (If you do `ps', the child will have a `Z' in its status field to indicate this.)

26.What is a shell?

A shell is an interactive user interface to an operating system services that allows an user to enter commands as character strings or through a graphical user interface. The shell converts them to system calls to the OS or forks off a process to execute the command. System call results and other information from the OS are presented to the user through an interactive interface. Commonly used shells are sh,csh,ks etc.