

COMPUTER SCIENCE

DSE-2 (UNIX SHELL PROGRAMMING)

FILL IN THE BLANKS

Introduction to Unix Operating Systems:

1. Unix is a _____ operating system known for its stability and robustness.
 - Answer: multi-user
2. Unix is designed to be highly _____.
 - Answer: portable
3. Unix was developed at _____ University in the late 1960s.
 - Answer: AT&T's Bell Labs
4. Unix uses a _____ interface for user interaction.
 - Answer: command-line
5. Name one key advantage of Unix over other operating systems.
 - Answer: Multitasking

Difference between Unix and other operating systems:

6. Unix is known for its _____ architecture, whereas some other operating systems use a monolithic kernel.
 - Answer: modular
7. Unix is POSIX compliant, which ensures _____ compatibility with other Unix-like operating systems.
 - Answer: binary
8. Unix emphasizes _____ and user-level customization.
 - Answer: flexibility
9. Most Unix-like operating systems are open-source, which means their source code is _____ to the public.
 - Answer: available
10. Unix systems use a hierarchical _____ system for organizing files and directories.
 - Answer: file

Features and Architecture:

11. Unix follows a _____ architecture, where multiple programs can run concurrently.
 - Answer: multi-process
12. The core of the Unix operating system is called the _____.
 - Answer: kernel
13. _____ is a Unix feature that allows users to schedule tasks at specific times.
 - Answer: Cron
14. _____ allows users to execute a command in the background and continue using the terminal.
 - Answer: Job control
15. Unix uses _____ as its primary programming language.
 - Answer: C

Installation, Booting, and Shutdown Process:

16. During the Unix installation process, users can typically choose their preferred _____.
 - Answer: software packages
17. The _____ process involves loading the Unix kernel into memory and initializing hardware.
 - Answer: booting

18. The command to initiate a system shutdown gracefully is _____.

- Answer: shutdown

19. To reboot a Unix system immediately, the command is _____.

- Answer: reboot

20. The init process, which has process ID 1, is responsible for system _____ and _____.

- Answer: initialization, shutdown

System Processes (an overview):

21. System processes are typically identified by their _____.

- Answer: process ID (PID)

22. The _____ process is the first process started by the kernel.

- Answer: init

23. The _____ process manages system resource allocation.

- Answer: scheduler

24. The _____ process handles disk I/O operations.

- Answer: swapper

25. The command to view a list of currently running processes is _____.

- Answer: ps

External and Internal Commands:

26. External commands are also known as _____ commands.

- Answer: executable

27. Internal commands are also known as _____ commands.

- Answer: built-in

28. The _____ command is used to display the manual pages for other commands.

- Answer: man

29. The _____ command is used to create a new directory.

- Answer: mkdir

30. To remove a file, the _____ command is used.

- Answer: rm

Creation of Partitions in OS:

31. Partitions in Unix are also known as _____.

- Answer: slices

32. The _____ command is commonly used to create and manage disk partitions.

- Answer: fdisk

33. Each partition in Unix is mounted to a specific _____ in the file system.

- Answer: directory

34. The _____ command is used to format a partition with a file system.

- Answer: mkfs

35. To view information about disk partitions, the command is _____.

- Answer: df

Processes and Its Creation Phases (Fork, Exec, Wait, Exit):

36. The _____ system call is used to create a new process in Unix.

- Answer: fork

37. After forking, the child process typically calls _____ to execute a different program.

- Answer: exec

38. The _____ system call is used by a parent process to wait for its child to terminate.

- Answer: wait

39. A process can exit using the _____ system call.

- Answer: exit

40. The exit status of a process is accessible using the _____ command.

- Answer: echo \$?

User Management and the File System:

41. Unix supports multiple types of users, including _____ and _____.

- Answer: regular users, superusers

42. The command to create a new user is _____.

- Answer: useradd

43. To grant additional rights to a user, you can use the _____ command.

- Answer: chmod

44. The _____ command is used to change a user's password.

- Answer: passwd

45. Unix supports user quotas to limit _____ usage.

- Answer: disk

File System Management and Layout:

46. The root directory in Unix is denoted by the symbol _____.

- Answer: /

47. The _____ directory contains system binaries and commands.

- Answer: /bin

48. User home directories are typically located in the _____ directory.

- Answer: /home

49. Configuration files are often found in the _____ directory.

- Answer: /etc

50. Temporary files are stored in the _____ directory.

- Answer: /tmp

File Permissions:

51. Unix file permissions are divided into three sets for _____ users, _____ users, and _____ users.

- Answer: owner, group, others

52. The _____ command is used to change file permissions.

- Answer: chmod

53. To see file permissions, the command is _____.

- Answer: ls -l

54. In a permission string like "rwxr-xr--," the first character represents the _____.

- Answer: file type

55. The "chmod 755 file.txt" command gives the owner _____ access.

- Answer: read, write, execute

Login Process:

56. Users can log in to a Unix system using a _____ and _____.

- Answer: username, password

57. The _____ file stores user account information, including login shell and home directory.

- Answer: /etc/passwd

58. The _____ file stores user password hashes.

- Answer: /etc/shadow

59. After successful login, users are typically placed in their _____ directory.

- Answer: home

60. The _____ file records login sessions and user commands.

- Answer: /var/log/auth.log

Managing Disk Quotas:

61. Disk quotas help prevent users from _____ disk space.

- Answer: exceeding

62. The _____ command is used to set disk quotas for users.

- Answer: setquota

63. To check disk usage and quotas, use the _____ command.

- Answer: quota

64. Users exceeding their disk quotas may receive _____ warnings.

- Answer: email

65. The _____ file keeps track of disk usage.

- Answer: /etc/mtab

Links (Hard Links, Symbolic Links):

66. A _____ link is a reference to the same inode as the original file.

- Answer: hard

67. A _____ link is a reference to the filename of the original file.

- Answer: symbolic

68. Hard links cannot link to _____ or directories.

- Answer: directories

69. Symbolic links can link to _____ and directories.

- Answer: files

70. The command to create a symbolic link is _____.

- Answer: ln -s

Shell Introduction and Shell Scripting:

71. The Unix shell is a _____-line interface for interacting with the operating system.

- Answer: command

72. Common Unix shells include Bash, _____, and Zsh.

- Answer: sh

73. The _____ editor is a commonly used text editor in Unix.

- Answer: vi

74. In vi editor, there are _____ modes: command mode, insert mode, and ex mode.

- Answer: three

75. A _____ is a script that contains a series of Unix commands.

- Answer: shell script

Writing and Executing Shell Scripts:

76. Shell scripts typically start with a _____ that specifies the shell to be used.

- Answer: shebang

77. To make a shell script executable, you need to set its _____ permissions.

- Answer: execute

78. The _____ command is used to execute a shell script.

- Answer: ./script.sh

79. Shell scripts can take _____ arguments from the command line.

- Answer: positional

80. Comments in a shell script are denoted by the symbol _____.

- Answer: #

Shell Variable (User Defined and System Variables):

81. _____ variables are defined and used by the user in shell scripts.

- Answer: User-defined

82. The _____ variable stores the exit status of the last command.

- Answer: \$? (dollar question mark)

83. The _____ variable stores the current user's home directory.

- Answer: \$HOME

84. The _____ variable contains the process ID of the current shell.

- Answer: \$\$ (dollar dollar)

85. The _____ variable holds the number of arguments passed to a script or function.

- Answer: \$# (dollar hash)

System Calls and Using System Calls:

86. System calls provide an interface between _____ and the kernel.

- Answer: user programs

87. The _____ system call is used to create a new process.

- Answer: fork

88. The _____ system call is used to open a file.

- Answer: open

89. The _____ system call is used to read from a file.

- Answer: read

90. The _____ system call is used to write to a file.

- Answer: write

Pipes and Filters:

91. _____ is a mechanism for connecting the output of one command to the input of another.

- Answer: Pipes

92. The _____ command is used to filter and display specific lines from a text file.

- Answer: grep

93. The _____ command is used to sort lines in a text file.

- Answer: sort

94. The _____ command is used to count lines, words, and characters in a text file.

- Answer: wc

95. The _____ command is used to concatenate and display files sequentially.

- Answer: cat

Unix Control Structures and Utilities:

96. The _____ statement is used for decision-making in shell scripts.

- Answer: if-else

97. The _____ statement is used for multiple conditional branches.

- Answer: switch

98. _____ are used in shell scripts for repetitive tasks.

- Answer: Loops

99. The _____ command is used to cut and extract portions of lines from files.

- Answer: cut

100. The _____ command is used to join lines from two or more files based on a common field.

- Answer: join

SHORT TYPE

Introduction: Unix Operating Systems

1. What is Unix?
 - Unix is a multi-user, multitasking operating system developed in the late 1960s at AT&T's Bell Labs.
2. Name some popular Unix-based operating systems.
 - Linux, macOS (formerly OS X), AIX, HP-UX, and Solaris are some examples.
3. What sets Unix apart from other operating systems?
 - Unix is known for its multi-user, multitasking capabilities, and its philosophy of "everything is a file."
4. What is the primary philosophy of Unix?
 - The Unix philosophy emphasizes the use of small, simple, and modular tools that can be combined to perform complex tasks.
5. How does Unix handle file systems?
 - Unix uses a hierarchical file system where files and directories are organized in a tree-like structure.

Installation, Booting, and Shutdown Process

6. How is Unix typically installed?
 - Unix can be installed from installation media or downloaded from the internet, and installation often involves partitioning the disk and configuring system settings.
7. Describe the booting process in Unix.
 - The booting process involves loading the operating system into memory from disk and initializing system processes. The bootloader plays a crucial role in this process.
8. How is a Unix system shutdown?
 - Unix systems can be shut down using commands like "shutdown" or "reboot," which initiate a controlled system shutdown.

System Processes

9. What are system processes?
 - System processes are essential background processes that manage various aspects of the operating system and are crucial for its proper functioning.
10. Give an example of a system process.
 - The init process (or its successor) is an example of a system process responsible for managing the system's startup and shutdown.

External and Internal Commands

11. What are external commands in Unix? - External commands are standalone programs that can be executed directly from the command line.
12. Provide an example of an internal command in Unix.
 - "cd" (change directory) is an example of an internal command that is built into the shell and doesn't require a separate program.

Creation of Partitions

13. Why might you need to create partitions in Unix? - Partitions help organize disk space and can be used to separate data or to install multiple operating systems on the same disk.
14. What tool can you use to create partitions in Unix?
 - Tools like "fdisk" or "parted" are commonly used to create and manage partitions.

Processes and Creation Phases

15. What are the main phases involved in creating a new process in Unix? - The main phases are Fork, Exec, Wait, and Exit.
16. Explain the "Fork" phase in process creation.
 - "Fork" creates a copy of the current process, resulting in two identical processes.
17. What is the purpose of the "Exec" phase in process creation?

- The "Exec" phase replaces the current process with a new program, effectively starting a different program within the same process ID (PID).

18. When is the "Wait" phase used in process creation?

- The "Wait" phase is used when one process needs to wait for the completion of another process before continuing.

19. What happens during the "Exit" phase in process creation?

- The "Exit" phase terminates a process and releases any associated resources.

User Management and File System

20. What are the types of users in Unix? - Unix typically has three types of users: superuser (root), system users, and regular users.

21. How can you create a new user in Unix?

- The "useradd" command is used to create new user accounts.

22. How do you grant rights to users in Unix?

- File permissions and ownership can be modified using commands like "chmod" and "chown."

23. Name some user management commands in Unix.

- "useradd," "userdel," "passwd," and "usermod" are common user management commands.

24. What is file quota in Unix?

- File quota restricts the amount of disk space a user or group can use on the file system.

25. What are some file systems available in Unix?

- Unix supports various file systems, including ext4, ZFS, and XFS.

26. Explain the concept of file permissions.

- File permissions determine who can read, write, or execute a file and are represented as a combination of letters or numbers.

27. Describe the login process in Unix.

- The login process involves authenticating a user, loading their environment, and providing access to the command shell.

28. How do you manage disk quotas in Unix?

- Disk quotas can be managed using commands like "quota" and "edquota" to set and monitor user quotas.

29. What are hard links and symbolic links in Unix?

- Hard links point directly to the same inode on disk, while symbolic links are pointers to file paths.

Shell Introduction and Shell Scripting

30. What is a shell in Unix? - A shell is a command-line interface that allows users to interact with the Unix operating system.

31. Name some common Unix shells.

- Common shells include "Bash," "Zsh," "Korn," and "Csh."

32. What are some editors available in Unix for text editing?

- Popular editors include "vi," "nano," "emacs," and "gedit."

33. How many modes of operation does the "vi" editor have?

- The "vi" editor has two main modes: command mode and insert mode.

34. What is a shell script?

- A shell script is a file containing a sequence of Unix commands that can be executed as a single unit.

35. How can you write and execute a shell script?

- Write the commands in a text file, make it executable using "chmod," and run it with "./scriptname."

36. What are shell variables in Unix?

- Shell variables are placeholders for data that can be used within shell scripts and commands.

37. Give an example of a system variable in Unix.

- "PATH" is a system variable that specifies directories to search for executable files.

38. What are system calls in Unix?

- System calls are low-level functions used to interact with the operating system kernel.

39. How are pipes and filters used in Unix?

- Pipes "|" are used to send the output of one command as input to another. Filters are commands that process or filter data.

Unix Control Structures and Utilities

40. How can you implement decision-making in shell scripts? - Decision-making can be implemented using "if-else" statements or "switch" statements in shell scripts.

41. What are loops in shell scripting?

- Loops are control structures that allow you to repeat a set of commands multiple times.

42. What are functions in Unix shell scripts?

- Functions are reusable code blocks that can be defined and called within a shell script.

43. Name some utility programs in Unix.

- Utility programs like "cut," "paste," "join," "tr," and "uniq" perform specific tasks on data.

44. What is the purpose of the "grep" utility?

- "grep" is used for pattern matching and searching text within files.

LONG TYPE

Introduction to Unix Operating Systems:

1. What is Unix, and how does it differ from other operating systems?
2. Explain the historical background of Unix.
3. Describe the key features of Unix operating systems.
4. What is the architecture of Unix? Explain its components.
5. Walk through the installation process of a Unix-based OS.
6. Explain the steps involved in the booting and shutdown process of a Unix system.
7. What are system processes, and how do they differ from user processes?

External and Internal Commands:

8. Differentiate between external and internal commands in Unix.
9. Provide examples of external and internal commands commonly used in Unix.
10. How can you create partitions in a Unix operating system?

Processes and Creation Phases:

11. Explain the concept of processes in Unix.
12. What are the phases involved in the creation of a process in Unix?
13. Describe the Fork system call and its purpose.
14. Explain the Exec system call in Unix.
15. What is the purpose of the Wait system call?
16. How does a process exit in Unix, and what is the significance of the Exit system call?

User Management and File System:

17. Describe the different types of users in a Unix system.
18. Walk through the process of creating a new user in Unix.
19. How can you grant rights and permissions to users in Unix?
20. List and explain common user management commands in Unix.
21. What is file quota, and how can it be set for users in Unix?
22. Discuss various file systems available in Unix.
23. Explain the management and layout of the Unix file system.
24. Describe the concept of file permissions in Unix.
25. Walk through the login process in Unix.
26. How can disk quotas be managed in Unix?
27. Explain the differences between hard links and symbolic links in Unix.

Shell Introduction and Shell Scripting:

28. What is a Unix shell, and why is it important?
29. Name various types of shells available in Unix.
30. Describe the key features of the Vi text editor.
31. Explain the different modes of operation in the Vi editor.
32. What is a shell script, and why is it used?
33. Provide a step-by-step guide on writing and executing a shell script in Unix.
34. Differentiate between user-defined and system variables in Unix.
35. How do system calls work in Unix, and why are they important?
36. Give examples of using system calls in Unix.
37. What are pipes and filters, and how are they used in Unix?

Unix Control Structures and Utilities:

38. Describe the decision-making structures available in Unix shell scripts (If-Else, Switch).
39. Explain how loops are implemented in Unix shell scripting.
40. What are functions, and how are they defined and used in shell scripts?
41. List and describe common utility programs in Unix (cut, paste, join, tr, uniq).
42. What is the purpose of the grep utility in Unix?
43. How does pattern matching work in Unix, and how is it used in grep?