Section 3: Shell Scripting

Questions

Please read the following questions carefully and try to answer them. You will find the correct answers, with an explanation where necessary, in the Answers.pdf file located in the same folder.

1. A shell script can automate repetitive work. It can also be scheduled using a cron jobs
    a. True
    b. False
2. Examples of Linux shells include (choose all that apply):
    a. BASH
    b. SH
    c. TCH
    d. KORN
3. If you are working in EMACS mode on a BASH shell, you use the following command to go the start of the current line
    a. CTRL-E
    b. CTRL-A
    c. CTRL-D
    d. CTRL-K
4. In EMACS mode, when you want to search your command history, you use the following command combination:
    a. CTRL-R
    b. CTRL-S
    c. /
    d. CTRL-O
5. To leave EMCS mode and move to vi mode while editing in a BASH shell, you issue the following command:
    a. set vi
    b. vi
    c. set –o vi
    d. set editor=vi
6. In vi mode, if you want to go to the end of the current line, you'd issue the following command:
    a. 0
    b. w
    c. h
    d. $
7. To search command history while using the vi mode in BASH, you issue the following command:
    a. CTRL-R

    b. /
    c. CTRL-S
    d. S
8. A given process contains only 3 file descriptors: STDIN, STDOUT, and STDERROR
    a. True
    b. False
9. You can examine different file descriptors of a process by navigating to the following directory:
    a. /proc
    b. /process
    c. /fd
    d. /tmp
10. The >> sign is used to redirect the standard output of a process to a file. The text inside this file will get:
    a. Overwritten
    b. Appended to
    c. Cleared
    d. Nothing will happen to the text
11. You can redirect standard input and standard error both to the same destination by using the following sign:
    a. &>
    b. >&
    c. &&
    d. >>
12. If you want to redirect the output of a given process to the standard input of another process, you use the following sign:
    a. >
    b. >>
    c. |
    d. %
13. Spaces are ignored when assigning variables in BASH.
    a. True
    b. False
14. The difference between single quotes and double quotes in BASH is:
    a. There is no difference
    b. Variables get resolved inside the double quotes but are output literally if in single quotes.
    c. BASH will work faster if you used single quotes
    d. Single quotes cannot be concatenated to other strings while double quotes can
15. The profile file of a given user is used to set environment variables, startup scripts, and any customizations needed for his/her session.
    a. True
    b. False

16. When using the cut command, you want to specify the delimiter that the command , you use the following command switch:
    a. –f
    b. –d
    c. –c
    d. –o
17. In the cut command, you must direct the file to the command's standard input using the < sign
    a. True
    b. False
18. If you are using the sort command, you want to use dictionary sorting rather than numerical one. You use the following command line switch:
    a. –n
    b. –c
    c. –b
    d. –d
19. While using the sort command, you can reverse order using the following switch:
    a. –t
    b. –g
    c. –r
    d. –e
20. The uniq command can take any form of lines and output the unique values
    a. True
    b. False
21. If you want to count the output of uniq command, either the duplicated lines or the unique ones, you use the following switch:
    a. –c
    b. –d
    c. –r
    d. You cannot count lines in uniq command
22. If you want to count lines using the wc command, you use the following switch:
    a. –l
    b. –c
    c. –w
    d. –count=lines
23. In the head command, you want to print all lines of a text file except the last 10 lines. You use the following form:
    a. head --lines=-10
    b. head --lines=10
    c. head -10
    d. head +10
24. You are using the tail command to monitor a log file, to which a specific process with PID 15654 is writing. You want to make sure that tail stops monitoring the file as soon as the process exits. You use the following command form:

a. tail –p 15654
b. tail pid=15654
c. tail --pid=15654
d. Just open another terminal and monitor the process using ps –ef

25. When using grep to filter the output of ps –ef searching for the occurrence of a specific process, you want to make sure that the grep command itself is not in the output. You use the following form of grep to do that:
    a. grep --ignore grep
    b. grep –v grep
    c. grep –l grep
    d. grep –c grep
26. You cannot use the commands you use in a BASH script separately on the command line
    a. True
    b. False
27. In BASH, you must use semicolons after each command
    a. True
    b. False
28. To execute a BASH script, you must start the script with #!/bin/bash
    a. True
    b. False
29. A shell script file must end in .sh extension or it won't get executed
    a. True
    b. False
30. The difference between pritnf and echo is that printf can handle special characters like \n and \t while echo cannot do that natively
    a. True
    b. False
31. The $0 special variable in BASH denotes the following:
    a. The first command line argument passed to the script
    b. The script name
    c. The shell name
    d. The username
32. If you want to determine the total number of command line arguments passed to a script, you use the following special variable:
    a. $#
    b. $0
    c. $COUNT
    d. $$
33. In BASH, a function must be called like file function_name().
    a. True
    b. False
34. If you want to make a local copy of a variable in BASH, you use the following command:
    a. private
    b. nonglobal

    c. local
    d. $
35. When you want to test two numbers for equality in a BASH script, you use the following command:
    a. ==
    b. eq
    c. -eq
    d. equal
36. It is wise to place a sleep interval in an infinite while loop to protect the server resources from over use
    a. True
    b. False
37. Perl uses the following to mark a code block:
    a. Do..done
    b. {..}
    c. Begin..end
    d. 4 space indentation
38. Perl statements cannot be used outside a perl script
    a. True
    b. False
39. In Perl, arrays are denoted by the following sign:
    a. $
    b. @
    c. %
    d. #
40. To get the total number of items in a Perl array @usernames, you use the following form:
    a. $#usernames
    b. #@usernames
    c. #$usernames
    d. $#usernames + 1
41. Perl can accept multiple lines of input from the user
    a. True
    b. False
42. To get the first command line argument passed to a Perl script, you use the following special variable:
    a. $ARGV[0]
    b. $0
    c. $1
    d. None of the above
43. In Perl, the following sign is used to test two strings for equality:
    a. –eq
    b. eq
    c. ==

    d. None of the above
44. Python requires you to add semicolons after each command:
    a. True
    b. False
45. The following Python module is used to accept arguments from the command line:
    a. sys
    b. os
    c. time
    d. date
46. The following function is used to accept input from a user in Python
    a. raw_input()
    b. input()
    c. read
    d. None of the above
47. Python recognizes a block of code when it starts with : and it indented 4 spaces to the right
    a. True
    b. False
48. There is no difference between a list and a tuple in Python
    a. True
    b. False
49. A dictionary is Python is like a hash in Perl
    a. True
    b. False
50. The following module in Python provides the sleep function, which pauses execution for a specified number of seconds:
    a. Time
    b. Sleep
    c. Date
    d. None of the above