

Linux Practical Assignment

Dr. J. V. Smart

Table of Contents

- [Linux Practical Problems](#)

IMPORTANT NOTE: Save all solutions in a file `PID-linux-assignment.txt` (where `PID` is your PID) in the following format

PID:

Name:

1.

command(s)

2.

command(s)

Notes

1. Write only the commands / contents of the script. Do not write the output
2. You are not required to validate the input provided by the user
3. You **must** solve the first 40 problems in order
4. For the rest of the problems, it is not necessary to solve the problems in any particular order. But in the assignment, write the solutions in order
5. Do not skip any number. If you have not solved a problem at the moment, write its number and leave a blank line
6. The assignment must be submitted in written or soft copy form
7. The assignment must be submitted before your internal practical examination

8. More problems may be added to this assignment

Linux Practical Problems

Write commands for the following (do not write the output)

1. Display the current directory
2. Go to your home directory
3. Create a directory called `dir1`
4. Go to the directory `dir1`
5. Create a file `a1.txt` and type some content in it
6. Display the contents of `a1.txt` on screen
7. Create a directory called `dir2`
8. Go to the directory `dir2`
9. create a file `a2.txt` and type some content in it
10. Go to your home directory
11. Display the contents of the file `a2.txt` using relative path
12. Display the contents of the file `a2.txt` using absolute path
13. Go to the directory `dir2`
14. Delete the file `a2.txt`
15. Go to the parent directory
16. Copy the file `a1.txt` to `a3.txt`
17. Rename the directory `dir2` to `dir3`
18. Rename the file `a3.txt` to `a4.txt`
19. Move the file `a4.txt` to `dir3`
20. Copy the file `a1.txt` to `dir3`
21. Go to the directory `dir3`

22. Display the contents of the current directory
23. Display the detailed list of the contents of the current directory
24. Go to your home directory
25. Display the detailed list of the contents of the directory `dir3` using relative path
26. Display the detailed list of the contents of the directory `dir3` using absolute path
27. Go to the directory `dir3`
28. Edit the file `a1.txt` and change the contents
29. Delete the file `a1.txt`
30. Go to the parent directory of the parent directory
31. Go to the directory `dir1`
32. Delete the directory `dir3`
33. Go to the parent directory
34. Create a directory `dir4`
35. Go into `dir4`
36. create a file `a5.txt` with some content
37. Display the current directory
38. Go to your home directory
39. Move the directory `dir4` in to `dir1`
40. Delete the directory `dir1` with all the contents
41. Create a variable `x`
42. Display the value of the variable `x`
43. Change the value of the variable `x` to `30`
44. Assign the value `7` to variable `y`

45. Display the values of both `x` and `y`
46. Store the sum of `x` and `y` in to the variable `z`
47. Store the subtraction of `x` and `y` in to the variable `z`
48. Store the multiplication of `x` and `y` in to the variable `z`
49. Store the division of `x` and `y` in to the variable `z`
50. Store the remainder of the division of `x` and `y` in to the variable `z`
51. Store the value of `z*(x+y)-5` in to the variable `a`
52. Display the message `a={value of a}` (display the actual value of `a`)
53. Display all files whose names begin with `b`
54. Display all files whose names begin with `b` and end in `n`
55. Display all files whose names begin with `b` , end in `n` and have exactly 3 characters in-between
56. Display all files whose names begin with 'asg'
57. Display all files whose names begin with 'asg' and end with `.txt`
58. Display all files whose names begin with 'asg' and have exactly two characters after that
59. Display all files whose names have exactly two characters followed by `.txt`
60. Assign the value `10` to `x` and `20` to `y`
61. If `x` is greater than `y` , then display 'Greater'
62. If `x` is less than `y` , then display 'Less'
63. If `x` is greater than or equal to `y` , then display 'Greater than or equal'
64. If `x` is less than or equal to `y` , then display 'Less than or equal'
65. If `x` is equal to `y` , then display 'Equal'
66. If `x` is not equal to `y` , then display 'Not equal'
67. Assign `abc` to `x`

68. Display `Zero length` if length of `x` is zero
69. Display `Non-zero length` if length of `x` is non-zero
70. Assign `abc` to `y`
71. Display `Equal` if `x` is equal to `y`
72. Display `Not equal` if `x` is not equal to `y`
73. Assign `querty` to `y`
74. Display `Equal` if `x` is equal to `y`
75. Display `Not equal` if `x` is not equal to `y`
76. Write a shell script to input the values of two numbers `n1` and `n2` . Display `Equal` if the numbers are equal, `Greater` if `n1>n2` and `Less` if `n1<n2`
77. Write a shell script to input the values of two numbers `n1` and `n2` . Display all the integers between `n1` and `n2` , including `n1` and `n2`
78. Write a shell script to process the natural numbers between `1` and `15` . If the number is divisible by 3, display the number followed by `divisible by 3` . If the number is not divisible by 3, display the number followed by `not divisible by 3` .
79. Write a shell script to accept a number `n` and display `n` lines of the following pattern:

```
*
**
***
****
*****
*****
```

80. Write a shell script to accept a number `n` and display `n` lines of the following pattern:

```
1
12
123
1234
12345
123456
```

81. Write a shell script to accept a number `n` and display `n` lines of the following pattern:

```
*****
*****
****
***
**
*

```

82. Write a shell script to accept a number `n` and display `n` lines of the following pattern:

```
  *
 *
*
 *
*
 *
*
```

83. Write a shell script to accept a number `n` and display `n` lines of the following pattern:

```
  *
 **
 ***
****
*****
*****
*****
```

84. Write a shell script to accept a number `n` and display `n` lines of the following pattern:

```
6
65
654
```

```
6543
65432
654321
```

85. Write a shell script to accept a number `n` and display `n` lines of the following pattern:

```
*      *
**     *
* *    *
*  *   *
*   *  *
*    **
*     *
```

86. Write a shell script to accept a number `n` and display `n` lines of the following pattern:

```
*          *
**         **
* *        * *
*  *       *  *
*   *      *   *
*    * *   *    *
*     *    *     *
```

87. Write a shell script to accept a number `n` and display $(n \times 2) + 1$ lines of the following pattern:

```
*      *
*      *
*      *
*
*      *
*      *
*      *
*      *
```

88. Write a shell script to accept a number `n` and display the first `n` terms of the following series:

```
1 3 5 7 9 11
```

89. Write a shell script to accept a number `n` and display the first `n` terms of

the Fibonacci series:

1 1 2 3 5 8 13 21 34

90. Write a shell script to accept two numbers `n1` and `n2` and display their total
91. Write a shell script to accept two numbers `n1` and `n2` and display their difference
92. Write a shell script to accept two numbers `n1` and `n2` and display their multiplication
93. Write a shell script to accept two numbers `n1` and `n2` and display their division
94. Write a shell script to accept two numbers `n1` and `n2` and display the remainder of their their division
95. Write a shell script to accept the principal amount `p` , the rate of interest `r` and the number of terms `n` and display the simple interest using the formula `p×r×n/100`
96. Write a shell script to accept two numbers `n1` and `n2` and an operator `op` (`op` may be `+` , `-` , `*` or `/`) and display the result of applying the operator to the numbers
97. Write a shell script to accept marks in 7 subjects and display their total
98. Write a shell script to accept marks in 7 subjects and display the percentage
99. Write a shell script to accept marks in 7 subjects and display `PASS` if the percentage is greater than or equal to 40 and `DETAINED` otherwise
100. Write a shell script to accept 7 numbers and display their total