

PRINT Name: _____ LAB Section:

Test Version: ____ One-Answer Multiple Choice 52 Questions – 25 of 25%

- ☞ Read **all** the words of these instructions and **both** sides (back and front) of all pages.
- ☞ Use your full, unabbreviated name on the mark-sense form. Do not abbreviate your name.
- ☞ Put the three-digit **Test Version** above into **NO. OF QUESTIONS** and **NO. OF STUDENTS**
- ☞ Fill in the bubbles with pencil only, no pen. Enter your NAME, Test Version, and answers.
- ☞ Manage your time. Answer questions you know, first. One Answer per question.
- ☞ The answer to the last question about reading/doing all these test instructions is: **Igen**

1. [27/113] If file **a** occupies one disk block, how many disk blocks are in use after this sequence of commands:

```
cp a b ; ln b c ; ln c d ; ln a e ; cp e d
```

a. 2 b. 4 c. 1 d. 3 e. 5
2. [34/113] In an empty directory, how many words are in file **a** after this:

```
echo It's redirected >b isn't it? >c ; ls >a
```

a. 2 b. 3 c. 0 d. 4 e. 1
3. [39/112] If files occupy one disk block, how many disk blocks will the system free up if I remove these four file names:

```
111 -rw-r--r-- 2 me me 1 Jan 1 1:00 a
111 -rw-r--r-- 2 me me 1 Jan 1 1:00 b
222 -rw-r--r-- 3 me me 1 Jan 1 1:00 c
222 -rw-r--r-- 3 me me 1 Jan 1 1:00 d
```

a. 3 b. 4 c. 2 d. 0 e. 1
4. [47/111] Dereference the following symlink **bar** into its equivalent absolute path:

```
mkdir -p /tmp/a/b ; ln -s ../../a../foo /tmp/a/b/bar
```

a. /tmp/a/foo b. /tmp/a/b/bar
c. /tmp/b/foo d. /tmp/foo
e. /tmp/b/bar

5. [48/113] If files occupy one disk block, how many disk blocks will the system free up if I remove these four file names:

```
111 -rw-r--r-- 1 me me 1 Jan 1 1:00 a
222 -rw-r--r-- 1 me me 1 Jan 1 1:00 b
333 -rw-r--r-- 1 me me 1 Jan 1 1:00 c
444 -rw-r--r-- 2 me me 1 Jan 1 1:00 d
```

a. 2 b. 0 c. 4 d. 3 e. 1
6. [53/112] How many arguments are passed to the command by the shell:

```
<foo foo -ils "-q"-z' >foo foo foo
```

a. 7 b. 4 c. 3 d. 5 e. 6
7. [55/113] If directory **/z** contains only these seven two-character names: **za**, **zb**, **zc**, **zd**, **z***, **z?**, **??**, then which command removes *only* the single two-character name **z?** from the directory?

a. `rm /z/z?` b. `rm /z/?\?` c. `rm "/z?"`
d. `rm /z\?` e. `rm '/z/z?'`
8. [56/113] If you are in **/etc** and `ls -l` shows a symbolic link **dir** `-> /foo` then dereference the absolute path of **dir** with no symbolic links:

a. `/foo` b. `/dir/foo`
c. `/etc/dir/foo` d. `/etc/foo`
e. `/etc/foo/dir`
9. [57/113] File **a** contains 2 lines. File **b** contains 3 lines. How many lines are in file **c** after this command line:

```
ln a e ; ln b d ; ln d c ; cat a b e d >c
```

a. 3 b. 0 c. 5 d. 4 e. 2
10. [59/112] If your **PATH** contained only the file names **/bin/sh**, **/bin/cat**, and **/bin/ls**, then what would be the output on your screen of this command: `cat /etc/passwd`

a. `bash: /bin/sh: command not found`
b. `cat: bash: no such file or directory`
c. `cat: /etc/passwd: command not found`
d. `bash: /bin/cat: no such file or directory`
e. `bash: cat: command not found`
11. [63/112] If file **a** occupies one disk block, how many disk blocks are in use after this sequence of commands:

```
ln a b ; ln b c ; cp c d ; ln c e ; rm a
```

a. 4 b. 5 c. 2 d. 1 e. 3

12. [64/113] How many arguments are passed to the command by the shell:
`foo It's "1 2" isn't it? I cannot decide.`
 a. 4 b. 2 c. 5 d. 6 e. 3
13. [68/113] What is the link count of directory **d** after these successful commands? `mkdir d d/a d/b d/b/z ; touch d/x d/y d/z`
 a. 6 b. 4 c. 5 d. 3 e. 2
14. [69/113] How many arguments are passed to the command by the shell:
`foo "bar "z " bin 'luk c' " wug'i "win' 9 8`
 a. 7 b. 11 c. 6 d. 4 e. 5
15. [69/112] File **a** contains 2 lines. File **b** contains 3 lines. How many lines are in file **c** after this command line:
`ln a d ; ln d c ; cat a b d >c`
 a. 0 b. 3 c. 5 d. 2 e. 4
16. [69/113] If file **a** occupies one disk block, how many disk blocks are in use after this sequence of commands:
`cp a b ; ln b c ; cp c d ; ln c e ; rm c`
 a. 4 b. 5 c. 1 d. 2 e. 3
17. [69/111] What displays on your screen given this command:
`date >date ; who >who ; head date | sort who`
 a. only the **who** displays because **sort** ignores the pipe
 b. **sort** reads the pipe and the **who** and displays both together
 c. nothing displays because **sort** ignores the pipe
 d. only the **date** displays because **sort** ignores the pipe
 e. **head** displays the **date** and **sort** displays the **who**
18. [70/113] If **/bin/foo** is a program that outputs **mom** and **/usr/bin/foo** is a program that outputs **dad** what would be the output on your screen of this two command sequence:
`PATH=/etc:/bin/foo:/usr/bin/foo:/usr ; foo`
 a. **dad**
 b. **bash: foo: command not found**
 c. **dad** followed by **mom**
 d. **mom** followed by **dad**
 e. **mom**

19. [71/112] What is the output on your screen after these command lines:
`echo 1 >x ; ln x y ; echo 2 >>y`
`cp y z ; ln z q ; head -1 x >y ; cat y`
 a. 1 followed by 2 b. 2 c. no output
 d. 1 e. 2 followed by 1
20. [72/113] If you are in **/etc** and `ls -l` shows a symbolic link **foo -> ../a/b** then dereference the absolute path of **foo** with no symbolic links:
 a. **/etc/foo/a/b** b. **/foo/./a/b**
 c. **/etc/a/b** d. **/etc/a/b/foo**
 e. **/a/b**
21. [73/112] If **/bin/foo** is a program that outputs **mom** and **/usr/bin/foo** is a program that outputs **dad**, what would be the output on your screen of this two command sequence:
`PATH=/etc:/dev:/usr/bin:/usr:/bin:/etc ; /bin/foo`
 a. **dad**
 b. **bash: /bin/foo: command not found**
 c. **dad** followed by **mom**
 d. **mom** followed by **dad**
 e. **mom**
22. [73/113] What is in file **c** after this command line:
`echo A >a ; ln a b ; echo B >b ; ln a c ; rm a b`
 a. nothing (empty file) b. **B**
 c. **A** d. no such file (nonexistent)
 e. **A** followed by **B**
23. [74/112] What is true about this output from `ls -il foo bar`
99 -rwxrwx--- 2 me me 3 Jan 1 1:00 foo
99 -rwx---rwx 2 me me 3 Jan 1 1:00 bar
 a. **foo** and **bar** are names for different files
 b. this output is not possible
 c. **foo** and **bar** are two of three names for the same file
 d. **foo** and **bar** each have three names (six names total)
 e. **foo** and **bar** are names for the same file
24. [75/113] How many arguments are passed to the command by the shell:
`<foo echo " 1 '2 3' 4 " 5 6 ' 7 "8 ' >out`
 a. 4 b. 5 c. 6 d. 2 e. 3

25. [78/112] If directory **a** contains only these five two-character names: **a?**, **11**, **?1**, **1***, **.1**, then which command removes *only* the single two-character name **?1** from the directory?
- a. `rm a/?1` b. `rm a/\??` c. `rm a/1*`
d. `rm a/*1` e. `rm a/??`
26. [79/113] If `/bin/bat` is a program that outputs **foo** and `/usr/bin/bat` is a program that outputs **bar** what would be the output on your screen of this two command sequence:
- ```
PATH=/etc:/usr:/usr/bin:/bin ; bat
```
- a. **foo** followed by **bar**  
b. **bar**  
c. **bar** followed by **foo**  
d. **foo**  
e. **bash: bat: command not found**
27. [80/113] Given this `ls -il` long listing:
- ```
234 drwxr-xr-x 567 me me 890 Jan 1 1:00 dir
```
- How many subdirectories lie immediately under **dir**?
- a. 234 b. 565 c. 567 d. 232 e. 888
28. [81/112] How many arguments are passed to the command by the shell:
- ```
echo It's not hard, it's just logical.
```
- a. 4                      b. 5                      c. 7                      d. 3                      e. 6
29. [81/113] If I have a directory named **a/b**, which action would increase its *link count* by exactly one?
- a. `touch a/b/c`                      b. `touch a/b2`                      c. `mkdir a/b2`  
d. `mkdir a/b/c`                      e. `ln a/b a/b2`
30. [81/112] What is in file **c** after this command line:
- ```
echo A >a ; ln a b ; echo B >>b ; ln a c ; rm a b
```
- a. no such file (nonexistent) b. **B**
c. **A** d. nothing (empty file)
e. **A** followed by **B**
31. [82/113] If file **a** occupies one disk block, how many disk blocks are in use after this sequence of commands:
- ```
cp a b ; ln b c ; cp c d ; cp c e
```
- a. 4                      b. 1                      c. 2                      d. 5                      e. 3

32. [83/112] What is the link count of directory **d** after these successful commands?
- ```
mkdir d ; cd d ; touch f g ; ln f a ; ln f b
```
- a. 4 b. 5 c. 3 d. 2 e. 1
33. [83/113] What is the link count of file **f** after these successful commands?
- ```
rm f ; touch f ; ln f g ; cp f f2
cp g a ; ln a b ; ln g c ; cp c a
```
- a. 1                      b. 5                      c. 2                      d. 4                      e. 3
34. [83/113] What is true about this output from `ls -il foo bar`
- ```
99 -r-x----- 2 me me 3 Jan 1 1:00 foo  
15 -r-x----- 2 me me 3 Jan 1 1:00 bar
```
- a. this output is not possible
b. **foo** and **bar** are names for the same file
c. **foo** and **bar** are two of three names for this file
d. **foo** and **bar** each have two names (four names total)
e. **foo** and **bar** each have three names (six names total)
35. [84/113] What is the link count of directory **d** after these successful commands?
- ```
mkdir d ; cd d ; touch a b ; mkdir 1 2 3
```
- a. 4                      b. 6                      c. 5                      d. 3                      e. 2
36. [84/113] What is the link count of file **f** after these successful commands?
- ```
rm f ; touch f ; cp f x ; cp f f2  
ln f a ; ln x y ; ln a z ; ln z q
```
- a. 4 b. 5 c. 3 d. 6 e. 2
37. [84/113] Which command usually goes in your `.bash_profile` file?
- a. `.bash_profile source`
b. `source ./bash_profile`
c. `.bashrc source`
d. `source ./bashrc`
e. `cat .bashrc`
38. [86/113] What is the link count of file **f** after these successful commands?
- ```
rm f ; touch f ; ln f b ; cp f g
cp b a ; ln a d ; ln b c ; cp c g
```
- a. 4                      b. 5                      c. 1                      d. 2                      e. 3

39. [86/113] What is the output on your screen after these command lines:  
`echo one >x ; ln x y ; echo ten >y`  
`echo one >>y ; echo two >x ; cat y`
- one followed by ten and two
  - no output on screen
  - one
  - two
  - ten
40. [87/113] The correct syntax to assign to a shell variable is:
- `var=some stuff`                      `b. var = some stuff`
  - `var = "some stuff"`                  `d. var="some stuff"`
  - `"var=some stuff"`
41. [89/113] What is the link count of file `f` after these successful commands?  
`rm f ; touch f ; ln f bar ; cp f f2`  
`cp bar x ; ln x y ; ln bar z ; ln z a`
- 2
  - 3
  - 5
  - 4
  - 1
42. [92/113] What command will recursively show disk usage in directories?
- `ls`
  - `df`
  - `tree`
  - `du`
  - `find`
43. [93/113] A "dangling symlink" is a symlink to:
- a special device file
  - a non-existent target
  - a directory
  - the current directory
  - a parent directory
44. [93/113] Which command shows the name of the current computer:
- `hostname`
  - `comname`
  - `whoami`
  - `history`
  - `find`
45. [93/112] Which of the following is true, given this long directory listing:  
`drwxr-x--x 12 me me 51 Jan 1 1:00 dir`
- The number 12 is the size of this directory.
  - The number 51 is the count of links (names) this directory has.
  - The number 12 is the count of links (names) this directory has.
  - The number 51 is the inode number of this directory.
  - The number 12 is the inode number of this directory.

46. [93/113] Which of the following is true, given this long directory listing:  
`drwxr-x--x 12 me me 51 Jan 1 1:00 dir`
- The number 51 is the inode number of this directory.
  - The number 12 is the inode number of this directory.
  - The number 12 is the size of this directory.
  - The number 51 is the count of links (names) this directory has.
  - The number 51 is the size of this directory.
47. [95/112] What is true about this output from `ls -il foo bar`  
`99 -r-x-----x 2 me me 3 Jan 1 1:00 foo`  
`99 -r-x-----x 2 me me 3 Jan 1 1:00 bar`
- this output is not possible
  - `foo` and `bar` are names for different files
  - `foo` and `bar` each have three names (six names total)
  - `foo` and `bar` are names for the same file
  - `foo` and `bar` are two of three names for this file
48. [96/112] Rewrite as a simplified absolute path (assume all directories exist):  
`/etc/me/../../you/../../etc/me/../../you/../../me/../../foo`
- `/etc/me/foo`
  - `/foo`
  - `/etc/foo`
  - `/etc/you/foo`
  - `/me/foo`
49. [97/113] How do you execute the program `bar` in the current directory?
- `bar/`
  - `$HOME/bar`
  - `./bar`
  - `bar/.`
  - `/bar`
50. [97/111] What is true about this output from `ls -il foo bar`  
`99 -r-x----- 2 me me 3 Jan 1 1:00 foo`  
`15 -r-x----- 2 me me 3 Jan 1 1:00 bar`
- `foo` and `bar` are names for the same file
  - `foo` and `bar` each have three names (six names total)
  - this output is not possible
  - `foo` and `bar` are two of three names for the same file
  - `foo` and `bar` are names for different files
51. [102/113] Rewrite as a simplified absolute path (assume all directories exist):  
`/lib/./bin/../../usr/../../etc/../../lib/./usr/../../bin/./bar`
- `/bar`
  - `/lib/bin/bar`
  - `/lib/usr/bar`
  - `/etc/bar`
  - `/lib/bar`

52. [107/110] **Did you read all the words of the test instructions on page one?**

- a. **Sim** (Yes - Portuguese)
- b. **Jes** (Yes - Esperanto)
- c. **Tak** (Yes - Polish)
- d. **Taip** (Yes - Lithuanian)
- e. **Igen** (Yes - Hungarian)

*This page intentionally left blank.*