

PRINT Name: _____ LAB Section:

Test Version: 229 One-Answer Multiple Choice 208 Questions – 15 of 15%

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

1. **Did you read all the words of the test instructions on page one?**
 - a. **Tak** (Yes - Polish)
 - b. **Jes** (Yes - Esperanto)
 - c. **Taip** (Yes - Lithuanian)
 - d. **Igen** (Yes - Hungarian)
 - e. **Sim** (Yes - Portuguese)
2. **My three-digit Lab Section number is:**
 - a. My lab room number, e.g. **B182, B119, J218, CA418**
 - b. My lecture section number, e.g. **010** or **020**.
 - c. The Test Version number printed in the top left corner.
 - d. My lecture room number, e.g. **T117**
 - e. The timetable section number of my weekly 2-hour lab period.
3. What displays on your screen given this command:
`date >date ; pwd >pwd ; head date | tail pwd`
 - a. only the **date** displays because **tail** ignores the pipe
 - b. only the **pwd** displays because **tail** ignores the pipe
 - c. **tail** reads the pipe and the **pwd** and displays both together
 - d. nothing displays because **tail** ignores the pipe
 - e. **head** displays the **date** and **tail** displays the **pwd**
4. Which of the following **PATH** statements makes the most sense?
 - a. **PATH=/bin:/usr/bin**
 - b. **PATH=/bin:/etc/passwd:/usr/bin**
 - c. **PATH=/bin/ls:/etc/passwd:/usr/bin**
 - d. **PATH=/bin/bash:/usr/bin:/bin**
 - e. **PATH=/bin:/usr/bin:/etc/passwd**

5. If **/bin/bat** is a program that outputs **foo** and **/usr/bin/bat** is a program that outputs **hi** what is the output on your screen after this:
`PATH=/usr:/usr/bin:/bin ; bat`
 - a. **hi** followed by **foo**
 - b. **hi**
 - c. **foo** followed by **hi**
 - d. **foo**
 - e. **bash: bat: command not found**
6. What is the link count of file **f** after these successful commands?
`rm f ; touch f ; cp f x
ln f a ; ln x y ; ln a z ; ln z q`
 - a. 2
 - b. 3
 - c. 4
 - d. 5
 - e. 6
7. What is the link count of directory **z** after these successful commands?
`mkdir z ; cd z ; touch a b ; mkdir c d e`
 - a. 4
 - b. 3
 - c. 5
 - d. 7
 - e. 6
8. File **a** contains 2 lines. File **b** contains 3 lines. How many lines are in file **c** after this:
`ln a d ; ln d c ; cat a b >c`
 - a. 5
 - b. 3
 - c. 0
 - d. 4
 - e. 2
9. How many arguments are passed to the command by the shell:
`<foo foo -x " " -z -r" " >foo 'foo foo'`
 - a. 6
 - b. 9
 - c. 7
 - d. 8
 - e. 5
10. In an empty directory, what is the output on your screen after this:
`touch 1 2 3 ; cow="*" ; echo "$cow"`
 - a. *****
 - b. **"\$cow"**
 - c. **\$cow**
 - d. **1 2 3**
 - e. **"1 2 3"**
11. How many files are touched? `touch '1 '2 3 '4' '5`
 - a. 2
 - b. 4
 - c. 3
 - d. 1
 - e. 5
12. How do you execute the program **foo** in the current directory?
 - a. **foo/.**
 - b. **./foo**
 - c. **\$HOME/foo**
 - d. **/foo**
 - e. **foo/**
13. If **/bin/foo** is a program that outputs **hi** and **/usr/bin/foo** is a program that outputs **mom** what is the output on your screen after this:
`PATH=/etc:/usr/bin:/bin ; foo`
 - a. **hi**
 - b. **mom**
 - c. **bash: foo: command not found**
 - d. **mom** followed by **hi**
 - e. **hi** followed by **mom**

14. If `/bin/bat` is a program that outputs `foo` and `/usr/bin/bat` is a program that outputs `bar` what is the output on your screen after this:
`PATH=/usr:/usr/bin:/bin ; bat`
- `foo` followed by `bar`
 - `bar`
 - `bash: bat: command not found`
 - `foo`
 - `bar` followed by `foo`
15. How many arguments are passed to the command by the shell:
`echo 'And it's not hard, it's just logical.'`
- 3
 - 4
 - 7
 - 6
 - 5
16. What command will recursively show disk usage in directories?
- `find`
 - `df`
 - `du`
 - `ls`
 - `tree`
17. What is true about this output from `ls -il foo bar`?
`15 -rwxrwxrwx 2 bin bin 3 Jul 31 12:33 foo`
`15 -rwxrwxrwx 3 bin bin 3 Jul 31 12:33 bar`
- `foo` and `bar` each have three names (six names total)
 - this output is not possible
 - `foo` and `bar` are names for the same file
 - `foo` and `bar` are two of three names for the same file
 - `foo` and `bar` are names for different files
18. The correct syntax to assign to a shell variable is:
- `V=foo bar`
 - `V = foo bar`
 - `"V=foo bar"`
 - `V="foo bar"`
 - `V = "foo bar"`
19. If `/bin/foo` is a program that outputs `mom` and `/usr/bin/foo` is a program that outputs `dad` what is the output on your screen after this:
`PATH=/bin/foo:/usr/bin/foo:/usr ; foo`
- `mom`
 - `dad`
 - `dad` followed by `mom`
 - `bash: foo: command not found`
 - `mom` followed by `dad`
20. What is the output on your screen after these command lines:
`echo one >x ; ln x y ; echo ten >y`
`echo two >x ; cat y`
- `two`
 - `one`
 - `ten`
 - `one` followed by `ten` and `two`
 - no output on screen
21. What is the output on your screen after this: `echo hi >out | wc -w`
- 2
 - 0
 - 3
 - 1
 - no output

22. What is true about this output from `ls -il foo bar`
`15 -r-x----- 2 me me 3 Jan 1 1:00 foo`
`15 -rwxrwxrwx 2 me me 3 Jan 1 1:00 bar`
- this output is not possible
 - `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 the same file
 - `foo` and `bar` are names for different files
23. If `/bin/prg` is a program that outputs `hi` and `/usr/bin/prg` is a program that outputs `foo` what is the output on your screen after this:
`PATH=/etc:/usr/bin:/bin ; prg`
- `hi` followed by `foo`
 - `bash: prg: command not found`
 - `foo`
 - `hi`
 - `foo` followed by `hi`
24. What is the output on your screen after this:
`echo 1 >x ; ln x y ; echo 2 >>y ; sort x`
- 2 followed by 1
 - 1 followed by 2
 - no output
 - 1
 - 2
25. 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 100 Jan 1 1:00 a`
`111 -rw-r--r-- 2 me me 100 Jan 1 1:00 b`
`222 -rw-r--r-- 3 me me 100 Jan 1 1:00 c`
`222 -rw-r--r-- 3 me me 100 Jan 1 1:00 d`
- 3
 - 2
 - 1
 - 0
 - 4
26. What is in file `foo` after this:
`echo hi >a ; ln a b ; echo me >b ; ln a foo ; rm a b`
- no such file (nonexistent)
 - nothing (empty file)
 - `hi` followed by `me`
 - `me`
 - `hi`
27. Dereference the following symlink `xyz` into its equivalent absolute path:
`ln -s ../../a/./b/./bar /tmp/a/b/xyz`
- `/tmp/bar`
 - `/tmp/b/bar`
 - `/tmp/a/bar`
 - `/tmp/b/xyz`
 - `/tmp/a/b/bar`

28. If `/bin/foo` is a program that outputs **one** and `/usr/bin/foo` is a program that outputs **two**, what is the output on your screen after this:
`PATH=/bin/ls:/home:/usr/bin/cat:/etc ; foo`
- two** followed by **one**
 - one** followed by **two**
 - bash: foo: command not found**
 - one**
 - two**
29. Which of these statements is true?
- Only single quotes are strong enough to stop GLOB patterns from expanding.
 - Only double quotes are strong enough to stop GLOB patterns from expanding.
 - If `/x` is an empty directory, `sort /x/*` produces an error message.
 - If `/y` is an empty directory, `echo /y/*` produces an error message.
 - Only backslashes are strong enough to stop GLOB patterns from expanding.
30. Which one of these names is usually a shell environment variable?
- foobar**
 - FOOBAR**
 - fooBar**
 - FooBar**
 - foobar**
31. If the file `pig` contained the word `bar`, what is the output on your screen after this: `PATH=/etc/passwd:/bin/ls:/bin/who ; /bin/cat pig`
- bar**
 - no output on screen
 - bash: /bin/cat: command not found**
 - pig**
 - /bin/cat: pig: No such file or directory**
32. Which of the following is true, given this long directory listing:
`drwxr-x--x 128 me me 32 Jan 1 1:00 dir`
- The number 128 is the inode number of this directory.
 - The number 32 is the count of links (names) this directory has.
 - The number 128 is the size of this directory.
 - The number 128 is the count of links (names) this directory has.
 - The number 32 is the inode number of this directory.
33. How many arguments are passed to the command by the shell:
`echo " 1 2 " three ' 4 ' five"6"`
- 4**
 - 1**
 - 9**
 - 5**
 - 3**
34. Which command line shows the current date?
- `echo date | bash`
 - `bash <date`
 - `date | bash`
 - `bash >date ; cat date`
 - `bash date`
35. In an empty directory, what is the output on your screen after this:
`echo hi >a ; ls | wc -w`
- 2**
 - no output
 - 1**
 - a**
 - 0**

36. What is in file `c` after this:
`echo A >a ; ln a b ; echo B >b ; ln a c ; rm a b`
- no such file (nonexistent)
 - A**
 - A** followed by **B**
 - nothing (empty file)
 - B**
37. Create a symbolic link under `/usr` named `bar` that has target `xy`:
- `ln -s 'xy' '/usr/bar'`
 - `ln -s '/usr/xy' /usr/bar`
 - `ln -s 'xy' /bar/usr`
 - `ln -s /usr/bar '/usr/xy'`
 - `ln -s /usr/bar 'xy'`
38. If `/bin/xxx` is a program that outputs **one** and `/usr/bin/xxx` is a program that outputs **two**, what is the output on your screen after this:
`PATH=/usr:/usr/bin:/etc:/bin ; xxx`
- two**
 - two** followed by **one**
 - one** followed by **two**
 - one**
 - bash: xxx: command not found**
39. 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 100 Jan 1 1:00 a
222 -rw-r--r-- 1 me me 100 Jan 1 1:00 b
333 -rw-r--r-- 2 me me 100 Jan 1 1:00 c
333 -rw-r--r-- 2 me me 100 Jan 1 1:00 d
```
- 4**
 - 1**
 - 0**
 - 2**
 - 3**
40. Given this `ls -il` long listing:
`123 drwxr-xr-x 456 me me 789 Jan 1 1:00 dir`
How many subdirectories lie immediately under `dir`?
- 123**
 - 454**
 - 789**
 - 787**
 - 456**
41. Which command removes *only* this five-character name containing a special character: `date`?
- `rm date/?`
 - `rm ./date?`
 - `rm ./date\?`
 - `rm date*`
 - `rm date\\?`
42. In an empty directory, what is the output on your screen after this:
`touch 1 2 3 ; cow="" ; echo '$cow'`
- `'$cow'`
 - `*`
 - `1 2 3`
 - `$cow`
 - `'1 2 3'`
43. How many arguments are passed to the command by the shell:
`<cow cow "-x "-y '-z' >cow cow`
- 3**
 - 7**
 - 5**
 - 6**
 - 4**

44. How many files are touched? `touch 1 "2 3" '4' 5`
 a. 5 b. 3 c. 7 d. 6 e. 4
45. How many files are touched? `touch "1" 2 3 " " 4 5`
 a. 7 b. 6 c. 5 d. 3 e. 4
46. Which command removes *only* this four-character name containing a special character: `?xyz`
 a. `rm '?xyz'` b. `rm '?xyz'` c. `rm ?xyz`
 d. `rm ''?xyz` e. `rm '?xyz'`
47. How many files are touched? `touch "1" " 2 3" " ' ' 4 5`
 a. 6 b. 3 c. 4 d. 7 e. 5
48. What is the link count of directory `d` after these successful commands?
`mkdir d ; mkdir d/a ; mkdir d/a/b ; mkdir d/a/c`
 a. 5 b. 4 c. 2 d. 1 e. 3
49. What displays on your screen given this command:
`ls >ls ; wc ls >wc ; sort ls | cat wc`
 a. `sort` displays the `ls` and `cat` displays the `wc`
 b. only the `wc` displays because `cat` ignores the pipe
 c. only the `ls` displays because `cat` ignores the pipe
 d. `cat` reads the pipe and the `wc` and displays both together
 e. nothing displays because `cat` ignores the pipe
50. What is the link count of file `f` after these successful commands?
`rm f ; touch f ; ln f a ; ln a b`
`cp f c ; ln c x ; rm b ; mv a b`
 a. 0 b. 4 c. 2 d. 3 e. 1
51. How many arguments are passed to the command by the shell:
`<wc wc " 1 '2 3' 4 " 5 6 ' 7 " 8 " ' >wc 9`
 a. 3 b. 4 c. 6 d. 5 e. 2
52. If you are in `/bin` and `ls -l` shows a symbolic link `bar -> ../dir/foo` then dereference the absolute path of `bar` with no symbolic links:
 a. `/bin/dir/foo/bar` b. `/dir/foo`
 c. `/bin/dir/foo` d. `/bin/bar/dir/foo`
 e. `/bar/../dir/foo`
53. 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. 1 b. 0 c. 4 d. 3 e. 2

54. In an empty directory, what is the output on your screen after this:
`touch A a ; echo * >"*" ; ls`
 a. `* A a` b. `A a >*` c. `A a >A a`
 d. `* >*` e. No output
55. Rewrite as a simplified absolute path:
`../../../../var/./a/../../../../var/b/../../../../etc/./bar/./foo`
 a. `/var/a/foo` b. `/etc/bar/foo` c. `/var/b/foo`
 d. `/etc/foo` e. `/var/foo`
56. If your terminal type is `xterm`, what is the output of this: `echo '$TERM'`
 a. `xterm` b. `$TERM`
 c. `'$TERM'` d. no output on screen
 e. `'xterm'`
57. In an empty directory, what is the output on your screen after this:
`echo one >.bar ; echo .*`
 a. `.bar`
 b. `one`
 c. `.*`
 d. `. .. .bar`
 e. an error message from `echo` saying `.*` does not exist
58. How many files are touched? `touch '1' "2 3 '4'" ' 5`
 a. 4 b. 3 c. 5 d. 2 e. 1
59. What is the link count of file `f` after these successful commands?
`rm f ; touch f ; ln f bar`
`cp bar x ; ln x y ; ln bar z ; ln z a`
 a. 5 b. 1 c. 4 d. 2 e. 3
60. If `/bin/pig` is a program that outputs `xx` and `/usr/bin/pig` is a program that outputs `foo` what is the output on your screen after this:
`PATH=/home:/bin:/dev:/usr/bin ; pig`
 a. `xx` followed by `foo`
 b. `foo` followed by `xx`
 c. `xx`
 d. `bash: pig: command not found`
 e. `foo`
61. File `a` contains 2 lines. File `b` contains 3 lines. How many lines are in file `c` after this: `sort a b >c ; cat a >>b ; cat c b >c a`
 a. 8 b. 5 c. 0 d. 12 e. 7
62. What is the link count of an empty directory?
 a. 2 b. 3 c. 0 d. 4 e. 1

63. If you want a user-defined alias in all your **bash** shells, what do you do?
- put the alias into the `/bin/bash` file for next log in
 - create the alias and then type **save** to save it to all shells
 - put the alias into the `/etc/passwd` file for next log in
 - put the alias into the `/etc/group` file for next log in
 - define the alias in my file `$HOME/.bashrc`
64. What is the link count of file **foo** after these successful commands?
- ```
rm foo ; touch foo ; ln foo bar ; ln bar x
cp bar a ; ln a b ; ln x c ; cp c d
```
- 5
  - 2
  - 3
  - 4
  - 1
65. What is the link count of directory **x** after these successful commands?
- ```
mkdir x ; mkdir x/y ; mkdir x/z ; mkdir x/y/z
```
- 2
 - 1
 - 3
 - 5
 - 4
66. If **foo** were a readable empty file, what is the output on your screen after this:
- ```
PATH=/etc/passwd:/bin/ls:/bin/cat ; /bin/cat foo
```
- `bash: cat: command not found`
  - `/bin/cat: foo: No such file or directory`
  - `bash: ls: command not found`
  - no output on screen
  - `bash: /bin/cat: command not found`
67. What is the link count of file **foo** after these successful commands?
- ```
rm foo ; touch foo ; ln foo bar
cp bar a ; ln a b ; ln bar c ; cp c a
```
- 3
 - 5
 - 1
 - 2
 - 4
68. What is the output on your screen after this:
- ```
mkdir foo ; rmdir foo | wc -w
```
- 3
  - 0
  - no output
  - 2
  - 1
69. File **a** contains 2 lines. File **b** contains 3 lines. How many lines are in file **c** after this: `ln a e ; ln b d ; ln d c ; cat e b >c`
- 4
  - 0
  - 5
  - 3
  - 2
70. Which command line allows programs in the current directory to execute without preceding the names with `./`? (P.S. Security Risk! Don't do this!)
- `$PATH=.:$HOME:/usr/bin`
  - `$PATH=/usr/bin:./bin`
  - `PATH=/bin:/usr/bin:.`
  - `PATH=./$HOME:/usr/bin`
  - `PATH=/usr/bin/.:$HOME`
71. How many arguments are passed to the command by the shell:
- ```
<bar bar -b "-a" '-r' >bar bar bar
```
- 5
 - 4
 - 7
 - 3
 - 6

72. What is usually in the environment variable `$SHELL`?
- the relative path of your login shell
 - the absolute path of your login shell
 - the absolute path of the system `/shell` directory
 - the relative path of the system `/shell` directory
 - the relative path of the `/home/shell` directory
73. What is in the local variable `$$` ?
- the first argument of the previous command line
 - the command name of the previous command line
 - `$$` is not a valid variable name
 - the cpu cost of the current session, in dollars
 - the process ID of the current shell
74. In an empty directory, what is the output on your screen after this:
- ```
echo one >.bar ; ls .????*
```
- an error message from `ls` saying `.????*` does not exist
  - `.bar`
  - `.????*`
  - `.. .bar`
  - `one`
75. The option to `ls` that shows inode (index) numbers is:
- `-x`
  - `-a`
  - `-l`
  - `-i`
  - `-1`
76. What is the link count of directory **dir** after these successful commands?
- ```
mkdir dir ; cd dir ; touch foo ; mkdir a b c
```
- 5
 - 2
 - 1
 - 4
 - 3
77. What is the link count of directory **dir** after these successful commands?
- ```
mkdir dir ; cd dir ; touch one ; mkdir two
```
- 1
  - 3
  - 2
  - 5
  - 4
78. If directory **dir** 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?
- `rm dir/?1`
  - `rm dir/\??`
  - `rm dir/1*`
  - `rm dir/??`
  - `rm dir/*1`
79. File **a** contains 2 lines. File **b** contains 3 lines. How many lines are in file **c** after this: `ln a d ; ln d e ; ln b f >c`
- 5
  - 0
  - 2
  - 3
  - 4
80. What is the link count of file **f** after these successful commands?
- ```
rm f ; touch f ; ln f b ; cp f c
cp b x ; ln x y ; ln b z ; ln z a
```
- 2
 - 3
 - 1
 - 4
 - 5

81. 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. 0            b. 1            c. 4            d. 3            e. 2
82. Which command line shows just the count of words in the file?
- a. `wc file | awk '{print #2}'`  
 b. `wc file | awk '[print $2]'`  
 c. `wc file | awk '[print #2]'`  
 d. `wc file | awk '{print $2}'`  
 e. `wc file | awk '{print 2}'`
83. What is the link count of directory `d` after these successful commands?
- ```
mkdir d ; cd d ; touch a ; mkdir b c
```
- a. 5 b. 3 c. 2 d. 6 e. 4
84. In an empty directory, what is the output on your screen after this:
- ```
touch a ; ls | wc -w
```
- a. 2            b. no output            c. 3  
 d. 0            e. 1
85. Which command line below shows only lines 6-10 of file `foo`?
- a. `head -6 foo | tail -10`            b. `tail -15 foo | head -5`  
 c. `head -10 foo | tail -6`            d. `head -10 foo | tail -5`  
 e. `tail -10 foo | head -6`
86. How many arguments are passed to the command by the shell:
- ```
echo "cow "y " bat 'man x' " pig'a "hop' a b
```
- a. 7 b. 11 c. 6 d. 4 e. 5
87. Which command removes *only* this four-character name containing a special character: `*xyz`
- a. `rm '*xyz` b. `rm '*xyz'` c. `rm "*xyz"`
 d. `rm "*"xyz"` e. `rm *xyz`
88. What is the output on your screen after this:
- ```
echo one >x ; ln x y ; echo two >>y ; sort x
```
- a. `two` followed by `one`            b. no output  
 c. `one`            d. `one` followed by `two`  
 e. `two`
89. How many arguments are passed to the command by the shell:
- ```
echo " one '2 three' 4 "five 6 ' 7 "8 ' >out
```
- a. 6 b. 4 c. 2 d. 3 e. 5

90. What is the output on your screen after this:
- ```
PATH=/bin/cat:/bin/sh:/bin/ls ; ls nosuchfile
```
- a. `ls: /bin/ls: command not found`  
 b. `bash: /bin/ls: command not found`  
 c. `ls: nosuchfile: No such file or directory`  
 d. `bash: /bin/sh: No such file or directory`  
 e. `bash: ls: command not found`
91. In an empty directory, what is the output on your screen after this:
- ```
echo one >.bar ; echo .??*
```
- a. an error message from `echo` saying `.??*` does not exist
 b. `.bar`
 c. `.. .bar`
 d. `.??*`
 e. `one`
92. You enter this `cp a/b c/` and get `cp: a: No such file or directory` because:
- a. you forgot to specify the destination file name after `c/`
 b. directory `a` does not exist
 c. pathname `a` exists but is a file, not a directory
 d. directory `c` does not exist
 e. the command `cp` is not in your search PATH
93. How many arguments are passed to the command by the shell:
- ```
<foo foo " a 'b c' d " e ' f " g " ' >foo
```
- a. 6            b. 5            c. 2            d. 4            e. 3
94. 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 100 Jan 1 1:00 a
222 -rw-r--r-- 2 me me 100 Jan 1 1:00 b
333 -rw-r--r-- 2 me me 100 Jan 1 1:00 c
444 -rw-r--r-- 1 me me 100 Jan 1 1:00 d
```
- a. 2 b. 1 c. 4 d. 0 e. 3
95. What is true about this output from `ls -il foo bar`
- ```
15 -r-x----- 2 me me 3 Jan 1 1:00 foo
99 -r-x----- 2 me me 3 Jan 1 1:00 bar
```
- a. this output is not possible  
 b. `foo` and `bar` each have three names (six names total)  
 c. `foo` and `bar` are two of three names for the same file  
 d. `foo` and `bar` are names for the same file  
 e. `foo` and `bar` are names for different files

96. What is the link count of directory **foo** after these successful commands?  
`mkdir foo ; cd foo ; touch a b c`  
 a. 3            b. 2            c. 1            d. 5            e. 4
97. If directory **/a** contains these seven two-character names: **aa, ab, ac, ad, a\*, a?, ??**, then which command removes *only* the single two-character name **a?** from the directory?  
 a. `rm "/a?"`            b. `rm '/a/a?'`            c. `rm /a\?`  
 d. `rm /a/a?`            e. `rm /a/?\?`
98. If directory **/a** contains these seven two-character names: **aa, ab, ac, ad, a\*, a., a..**, then which command removes *only* the single two-character name **a\*** from the directory?  
 a. `rm /a*`            b. `rm /a/a\*`            c. `rm /a/a?`  
 d. `rm /a/a*`            e. `rm /a/*`
99. How many arguments are passed to the command by the shell:  
`echo 'It's "1 2" isn't it? I can't decide.'`  
 a. 2            b. 5            c. 6            d. 3            e. 4
100. 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 100 Jan 1 1:00 a`  
`222 -rw-r--r-- 2 me me 100 Jan 1 1:00 b`  
`333 -rw-r--r-- 2 me me 100 Jan 1 1:00 c`  
`444 -rw-r--r-- 2 me me 100 Jan 1 1:00 d`  
 a. 0            b. 1            c. 3            d. 4            e. 2
101. What is the link count of file **f** after these successful commands?  
`rm f ; touch f ; ln f bar`  
`cp bar x ; ln x y ; ln bar z`  
 a. 5            b. 2            c. 4            d. 3            e. 1
102. File **a** contains 2 lines. File **b** contains 3 lines. How many lines are output on your screen by this: `cat b | cat a`  
 a. 3 followed by 2            b. 5            c. 2            d. 3            e. 2 followed by 3
103. Which command line would show the index (inode) number of a file?  
 a. `find -i file`            b. `cat -l file`            c. `ls -i file`  
 d. `ls -l file`            e. `cat -i file`
104. Rewrite as a simplified absolute path:  
`/home/me/./you/./././etc/././home/me/./you/./me/./foo`  
 a. `/home/foo`            b. `/home/me/foo`  
 c. `/etc/foo`            d. `/home/you/foo`  
 e. `/foo`

105. What is in file **c** after this:  
`echo B >b ; ln b a ; echo A >a ; ln a c ; rm a b`  
 a. no such file (nonexistent)            b. **A**  
 c. nothing (empty file)            d. **B**  
 e. **A** followed by **B**
106. What is in file **c** after this:  
`echo foo >a ; ln a b ; echo bar >>b ; ln a c ; rm a`  
 a. nothing (empty file)            b. **foo** followed by **bar**  
 c. no such file (nonexistent)            d. **foo**  
 e. **bar**
107. If `/bin/xxx` is a program that outputs **one** and `/usr/bin/xxx` is a program that outputs **two**, what is the output on your screen after this:  
`PATH=/bin/xxx:/usr/bin/xxx:/etc/passwd ; xxx`  
 a. **one** followed by **two**  
 b. **two** followed by **one**  
 c. **bash: xxx: command not found**  
 d. **one**  
 e. **two**
108. If your **PATH** variable contains `/bin:/usr/bin`, what is the output of this:  
`echo '$PATH'`  
 a. `'/bin:/usr/bin'`  
 b. `'$PATH'`  
 c. `$PATH`  
 d. **echo: \$PATH: No such file or directory**  
 e. `/bin:/usr/bin`
109. What is the link count of file **foo** after these successful commands?  
`rm foo ; touch foo ; ln foo bar`  
`cp bar x ; ln x y ; ln y z`  
 a. 2            b. 4            c. 1            d. 3            e. 0
110. 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 100 Jan 1 1:00 a`  
`111 -rw-r--r-- 2 me me 100 Jan 1 1:00 b`  
`222 -rw-r--r-- 2 me me 100 Jan 1 1:00 c`  
`222 -rw-r--r-- 2 me me 100 Jan 1 1:00 d`  
 a. 2            b. 0            c. 4            d. 3            e. 1
111. A "dangling symlink" is a symlink to:  
 a. a directory            b. a non-existent target  
 c. the current directory            d. a parent directory  
 e. a special device file

112. In an empty directory, how many words are in file **a** after this:  
`echo It's redirected >b isn't it\? ; ls >a`  
 a. 2            b. 1            c. 3            d. 4            e. 0
113. If you are in `/etc` and `ls -l` shows a symbolic link **bar** `-> ../foo` then dereference the absolute path of **bar** with no symbolic links:  
 a. `/bar/foo`            b. `/etc/foo/bar`            c. `/etc/foo`  
 d. `/etc/bar/foo`            e. `/foo`
114. If I have a directory named `/1/2`, which action would increase its *link count* by exactly one?  
 a. create a directory named `/1/2/3`  
 b. create a directory named `/1/2`  
 c. create one file named `/1/22`  
 d. create one file named `/1/2/3`  
 e. create a directory named `/1/22`
115. Which of the following is true, given this long directory listing:  
`drwxr-x--x 128 me me 32 Jan 1 1:00 dir`  
 a. The number 128 is the inode number of this directory.  
 b. The number 32 is the size of this directory.  
 c. The number 128 is the size of this directory.  
 d. The number 32 is the inode number of this directory.  
 e. The number 32 is the count of links (names) this directory has.
116. Rewrite as a simplified absolute path:  
`/usr/./bin/./lib/./lib/./lib/./etc/./usr/./lib/./bin/./bar`  
 a. `/etc/bar`            b. `/usr/lib/bar`            c. `/usr/bin/bar`  
 d. `/bar`            e. `/usr/bar`
117. What is the link count of directory **dir** after these successful commands?  
`mkdir dir ; mkdir dir/foo ; touch dir/bar`  
 a. 2            b. 5            c. 1            d. 3            e. 4
118. If `/bin/foo` is a program that outputs **dad** and `/usr/bin/foo` is a program that outputs **mom** what is the output on your screen after this:  
`PATH=/usr:/etc:/bin:/usr/bin ; foo`  
 a. **dad** followed by **mom**  
 b. **bash: foo: command not found**  
 c. **mom**  
 d. **mom** followed by **dad**  
 e. **dad**

119. What is true about this output from `ls -il foo bar`  
`15 -r-x-----x 2 me me 3 Jan 1 1:00 foo`  
`15 -r-x-----x 2 me me 3 Jan 1 1:00 bar`  
 a. this output is not possible  
 b. **foo** and **bar** are two of three names for this file  
 c. **foo** and **bar** each have three names (six names total)  
 d. **foo** and **bar** are names for different files  
 e. **foo** and **bar** are names for the same file
120. In an empty directory, what is in file **out** after this:  
`ls nosuchfile | wc -w >out`  
 a. nothing (empty file)            b. 0  
 c. **out**            d. 1  
 e. **nosuchfile**
121. Which command shows the name of the current computer:  
 a. **comname**            b. **whois**            c. **find**  
 d. **history**            e. **hostname**
122. If `/bin/foo` is a program that outputs **mom** and `/usr/bin/foo` is a program that outputs **dad**, what is the output on your screen after this:  
`PATH=/dev:/usr/bin:/usr:/bin:/etc ; /bin/foo`  
 a. **dad**  
 b. **mom**  
 c. **dad** followed by **mom**  
 d. **mom** followed by **dad**  
 e. **bash: /bin/foo: command not found**
123. What is the link count of directory **dir** after these successful commands?  
`mkdir dir ; cd dir ; touch a b c ; mkdir d e`  
 a. 7            b. 2            c. 5            d. 3            e. 4
124. In an empty directory, what is in file **count** after this:  
`ls ??? | wc -w >count`  
 a. 1 1 2            b. 1 1 1  
 c. 0            d. nothing (empty file)  
 e. 1
125. What is the link count of directory **z** after these successful commands?  
`mkdir z ; mkdir z/a ; touch z/b z/c z/d`  
 a. 1            b. 3            c. 4            d. 2            e. 5
126. How many arguments are passed to the command by the shell:  
`echo ' one two ' three ' four ' 5'6'`  
 a. 5            b. 6            c. 9            d. 4            e. 1
127. File **a** contains 2 lines. File **b** contains 3 lines. How many lines are in file **e** after this: `ln a d ; cp a f ; ln d c ; ln c e ; cat a b d f >e`  
 a. 3            b. 2            c. 9            d. 5            e. 6

128. Which command line outputs inode/filename pairs for names in the current directory, sorted by inode number?
- a. `sort -n | ls -ai`                      b. `ls -ai | sort -n`  
 c. `ls /* | sort -node`                    d. `ls -node * > sort -n`  
 e. `ls -i * > sort -n`
129. In an empty directory, what is the output on your screen after this:  
`echo one >.bar ; echo .?*`
- a. `.bar`  
 b. an error message from `echo` saying `.?* does not exist`  
 c. `one`  
 d. `.. .bar`  
 e. `.?*`
130. If file `one` occupies one disk block, how many disk blocks are in use after this:  
`cp one foo ; ln foo two ; ln two bar ; ln one cow`
- a. 1                      b. 4                      c. 5                      d. 3                      e. 2
131. How many arguments are passed to the command by the shell:  
`echo " 1 '2 3' 4 "5 6 ' 7 "8 ' >out`
- a. 5                      b. 2                      c. 6                      d. 3                      e. 4
132. What is the link count of directory `z` after these successful commands?  
`mkdir z ; cd z ; touch a ; ln a b ; ln a c`
- a. 2                      b. 3                      c. 1                      d. 5                      e. 4
133. In an empty directory, what is the output on your screen after this:  
`echo one >.bar ; echo .????*`
- a. an error message from `echo` saying `.????* does not exist`  
 b. `.. .bar`  
 c. `one`  
 d. `.????*`  
 e. `.bar`
134. What is the output on your screen after these command lines:  
`echo one >x ; ln x y ; echo two >>y`  
`sort x >y ; cat y`
- a. `one` followed by `two`                      b. `two`  
 c. no output                                      d. `two` followed by `one`  
 e. `one`
135. Which command line always prints just the two characters `$x` on the screen?
- a. `echo '$x'`                      b. `echo "$$x"`                      c. `echo $$x`  
 d. `echo "$x"`                      e. `echo $x`

136. 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 100 Jan 1 1:00 a
222 -rw-r--r-- 1 me me 100 Jan 1 1:00 b
333 -rw-r--r-- 1 me me 100 Jan 1 1:00 c
444 -rw-r--r-- 1 me me 100 Jan 1 1:00 d
```
- a. 3 b. 4 c. 0 d. 2 e. 1
137. What does *quoting* mean on a shell command line?
- a. typing a "control" character using the [CTRL] key
 b. setting the `PS1` variable to be your shell prompt
 c. turning off the special meaning of shell meta-characters
 d. using a leading tilde ("~") on a pathname to mean your `HOME` directory
 e. using more than one pathname argument to a command, e.g. `rm a b c`
138. In an empty directory, what is the output on your screen after this:
`touch 1 2 3 ; cow="*" ; echo "$cow"`
- a. "1 2 3" b. "\$cow" c. 1 2 3
 d. \$cow e. *
139. What is the link count of file `f` after these successful commands?
`rm f ; touch f ; cp f x`
`ln f a ; ln x y ; ln a z ; ln x b`
- a. 3 b. 6 c. 4 d. 2 e. 5
140. How many arguments are passed to the command by the shell:
`echo 'It's a bird! No! It's a plane!'`
- a. 5 b. 4 c. 1 d. 2 e. 3
141. File `a` contains 2 lines. File `b` contains 3 lines. How many lines are in file `c` after this: `ln a e ; ln b d ; ln d c ; cp d e ; sort a b e d >c`
- a. 4 b. 12 c. 6 d. 10 e. 7
142. How many files are touched? `touch 1 "2 3" ' ' 4 5`
- a. 5 b. 4 c. 3 d. 6 e. 7
143. If you are in `/bin` and `ls -l` shows a symbolic link `foo -> /bar` then dereference the absolute path of `foo` with no symbolic links:
- a. `/bin/bar` b. `/foo/bar` c. `/bin/bar/foo`
 d. `/bin/foo/bar` e. `/bar`
144. What is the output on your screen after these command lines:
`echo one >x ; ln x y ; echo two >y`
`echo ten >x ; cat y`
- a. `one` followed by `two` and `ten` b. no output on screen
 c. `one` d. `two`
 e. `ten`
145. What is the link count of directory `z` after these successful commands?
`mkdir z ; mkdir z/a z/a/b z/a/c z/a/d`
- a. 1 b. 5 c. 4 d. 3 e. 2

146. How many arguments are passed to the command by the shell:
`echo " 1 2 "three ' 4 ' five"6"`
 a. 3 b. 9 c. 1 d. 5 e. 4
147. What is the link count of directory **d** after these successful commands?
`mkdir d ; cd d ; touch f ; ln f a ; ln f b`
 a. 2 b. 4 c. 3 d. 1 e. 5
148. If the file **bat** contained the word **foo**, what is the output on your screen after this: `PATH=/etc/passwd:/bin/ls:/bin/cat ; /bin/ls bat`
 a. no output on screen
 b. `/bin/ls: bat: No such file or directory`
 c. `bash: /bin/ls: command not found`
 d. `bat`
 e. `foo`
149. If your **PATH** contained only the file names `/bin/sh`, `/bin/cat`, and `/bin/ls`, then what is the output on your screen of this command:
`cat /etc/passwd`
 a. `cat: /etc/passwd: command not found`
 b. `bash: /bin/cat: no such file or directory`
 c. `bash: cat: command not found`
 d. `cat: bash: no such file or directory`
 e. `bash: /bin/sh: command not found`
150. How many arguments are passed to the command by the shell:
`<foo foo " a 'b c' d " e ' f " g " ' >foo h`
 a. 2 b. 3 c. 4 d. 6 e. 5
151. If file **foo** occupies one disk block, how many disk blocks are in use after this:
`cp foo bar ; ln bar one ; cp one two ; cp one xxx`
 a. 5 b. 4 c. 1 d. 3 e. 2
152. 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 100 Jan 1 1:00 a`
`222 -rw-r--r-- 3 me me 100 Jan 1 1:00 b`
`222 -rw-r--r-- 3 me me 100 Jan 1 1:00 c`
`222 -rw-r--r-- 3 me me 100 Jan 1 1:00 d`
 a. 4 b. 3 c. 1 d. 2 e. 0
153. What is the link count of directory **dir** after these successful commands?
`mkdir dir ; touch foo ; cd dir ; ln ../foo bar`
 a. 5 b. 1 c. 3 d. 4 e. 2

154. What is true about this output from `ls -il foo bar`
`35 -rw-rw-r-- 2 me me 3 Jan 1 1:00 foo`
`36 -rw-rw-r-- 2 me me 3 Jan 1 1:00 bar`
 a. **foo** and **bar** each have two names (four names total)
 b. this output is not possible
 c. **foo** and **bar** are two of three names for this file
 d. **foo** and **bar** are names for the same file
 e. **foo** and **bar** each have three names (six names total)
155. What is usually in the environment variable **\$HOME**?
 a. the relative path of your login home directory
 b. the relative path of the system `/home` directory
 c. the absolute path of the system `/home` directory
 d. the absolute path of your login home directory
 e. the relative path of the `ROOT` directory
156. What is the output on your screen after this:
`echo hi >a ; cp a b | wc -w`
 a. 0 b. no output c. 3
 d. 2 e. 1
157. How many arguments are passed to the command by the shell:
`<bat bat -b "-a -r" >bat bat bat`
 a. 3 b. 7 c. 4 d. 6 e. 5
158. 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 100 Jan 1 1:00 a`
`222 -rw-r--r-- 1 me me 100 Jan 1 1:00 b`
`444 -rw-r--r-- 2 me me 100 Jan 1 1:00 c`
`444 -rw-r--r-- 2 me me 100 Jan 1 1:00 d`
 a. 2 b. 1 c. 4 d. 0 e. 3
159. Which option to **ls** displays the directory itself and not its contents?
 a. `-a` b. `-i` c. `-d` d. `-l` e. `-R`
160. How many files are touched? `touch 1 "2 3 ' 4 '" 5`
 a. 1 b. 5 c. 3 d. 2 e. 4
161. How many arguments are passed to the command by the shell:
`<bar bar -b"-a '-r' >bar" bar >out`
 a. 2 b. 6 c. 4 d. 3 e. 5
162. What is the link count of file **a** after these successful commands?
`ln a d ; cp a f ; ln d c ; ln f g ; ln c e`
 a. 1 b. 3 c. 2 d. 4 e. 5
163. How many arguments are passed to the command by the shell:
`<pig pig -x " " -z -r" " >pig pig pig`
 a. 6 b. 7 c. 8 d. 5 e. 9

164. If you are in `/bin` and `ls -l` shows a symbolic link `foo -> dir/bar` then dereference the absolute path of `foo` with no symbolic links:
- `/foo/dir/bar`
 - `/bin/foo/dir/bar`
 - `/bin/dir/bar/foo`
 - `/dir/bar`
 - `/bin/dir/bar`
165. What is the output on your screen after these command lines:
`echo 1 >x ; ln x y ; echo 2 >>y`
`head -1 x >y ; cat y`
- no output
 - 1
 - 1 followed by 2
 - 2
 - 2 followed by 1
166. If `/bin/foo` is a program that outputs `one` and `/usr/bin/foo` is a program that outputs `two`, what is the output on your screen after this:
`PATH=/etc:/usr/bin:/usr:/bin:/dev ; foo`
- one followed by two
 - two followed by one
 - one
 - `bash: foo: command not found`
 - two
167. What is the resulting link count of empty directory `dir` after these successful commands? `cd dir ; touch foo ; ln foo one ; ln foo two`
- 1
 - 4
 - 5
 - 2
 - 3
168. How many arguments are passed to the command by the shell:
`<cow cow "-x" -y '-z' >cow cow`
- 3
 - 6
 - 4
 - 2
 - 5
169. What is the output on your screen of this unquoted command line:
`mkdir a ; touch b a/b1 a/b2 ; find a -name b*`
- `b a/b1 a/b2`
 - `b`
 - `a/b1 a/b2`
 - `b1 b2`
 - no output
170. In an empty directory, what is the output on your screen after this:
`touch 1 2 3 ; cow="*" ; echo $cow`
- `"1 2 3"`
 - `1 2 3`
 - `"*"`
 - `*`
 - `$cow`
171. If `/bin/pig` is a program that outputs `hi` and `/usr/bin/pig` is a program that outputs `foo` what is the output on your screen after this:
`PATH=/etc:/usr/bin:/bin ; pig`
- `hi` followed by `foo`
 - `hi`
 - `foo` followed by `hi`
 - `bash: pig: command not found`
 - `foo`

172. How many arguments are passed to the command by the shell:
`echo 'It's a bird! It's a plane!'`
- 2
 - 3
 - 5
 - 1
 - 4
173. If directory `/a` contains these seven two-character names: `aa, ab, ac, ad, a?, a*, a.`, then which command removes *only* the single two-character name `a*` from the directory?
- `rm /a/a*`
 - `rm /a*`
 - `rm /a/*`
 - `rm /a/a?`
 - `rm "/a/a*"`
174. If `/bin/foo` is a program that outputs `one` and `/usr/bin/foo` is a program that outputs `two`, what is the output on your screen after this:
`PATH=/dev:/usr/bin:/usr:/bin:/etc ; /bin/foo`
- one
 - `bash: /bin/foo: command not found`
 - two followed by one
 - one followed by two
 - two
175. If the file `bat` contained the word `foo`, what is the output on your screen after this: `PATH=/bin/cat:/bin/who:/bin/ls ; cat bat`
- `bash: cat: command not found`
 - `bat`
 - no output on screen
 - `cat: bat: No such file or directory`
 - `foo`
176. If I have a directory named `a/b`, which action would increase its *link count* by exactly one?
- create a hard link to directory `b` named `b2`
 - create a directory named `a/b/c`
 - create a directory named `a/b2`
 - create a file named `a/b/c`
 - create a file named `a/b2`
177. File `a` contains 3 lines. File `b` contains 4 lines. How many lines are output on your screen by this: `sort a | echo b`
- 1
 - 3
 - 3 followed by 1
 - 3 followed by 4
 - 4
178. If you are in `/etc` and `ls -l` shows a symbolic link `bar -> foo` then dereference the absolute path of `bar` with no symbolic links:
- `/etc/bar/foo`
 - `/etc/foo`
 - `/foo`
 - `/etc/foo/bar`
 - `/bar/foo`
179. In an empty directory, what is the output on your screen after this:
`touch A a ; echo * ">*"`
- `A a`
 - No output
 - `A a >*`
 - `* >*`
 - `A a >A a`

199. Which command line makes pathnames `/usr/local/bin` and `/usr/bin` lead to the same directory?
- `ln . /usr/local`
 - `mkdir /usr/local`
 - `touch /usr/local`
 - `ln -s . /usr/local`
 - `rmdir /usr/local`
200. What is the link count of file `foo` after these successful commands?
- ```
rm foo ; touch foo ; ln foo bar
cp bar x ; ln x y ; ln bar z
```
- 4
  - 5
  - 1
  - 2
  - 3
201. What is usually in the environment variable `$PATH`?
- the absolute path of your login home directory
  - the absolute path of your login shell
  - a colon-separated list of your `passwd` file fields
  - the absolute path of the system `/path` directory
  - a colon-separated list of directories containing command names
202. What is the link count of file `foo` after these successful commands?
- ```
rm foo ; touch foo ; ln foo bar
cp bar x ; ln x y ; ln bar z ; ln z a
```
- 2
 - 1
 - 4
 - 3
 - 5
203. File `a` contains 2 lines. File `b` contains 3 lines. How many lines are in file `c` after this: `ln a d ; ln d c ; cp c b ; sort a b d >c`
- 4
 - 2
 - 5
 - 6
 - 0
204. Which command usually goes in your `.bash_profile` file?
- `.bashrc source`
 - `.bash_profile source`
 - `cat .bashrc`
 - `source ./bash_profile`
 - `source ./bashrc`
205. 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
```
- 4
  - 2
  - 5
  - 1
  - 3
206. What is the link count of directory `d` after these successful commands?
- ```
mkdir d ; mkdir d/a d/b ; touch d/c d/e
```
- 3
 - 1
 - 2
 - 4
 - 5
207. What is the link count of file `f` after these successful commands?
- ```
rm f ; touch f ; ln f bar
cp bar x ; ln x y ; ln y z
```
- 0
  - 2
  - 3
  - 4
  - 1
208. Did you read all the words of the test instructions on page one?
- Taip** (*Yes - Lithuanian*)
  - Sim** (*Yes - Portuguese*)
  - Tak** (*Yes - Polish*)
  - Igen** (*Yes - Hungarian*)
  - Jes** (*Yes - Esperanto*)

*This page intentionally left blank.*