

Evaluation: 57 Questions

Name: \_\_\_\_\_

**Important Instructions**

1. Read all the instructions and both sides (back and front) of all pages.
2. Manage your time when answering questions on this test.  
*Answer the questions you know, first.*

**Multiple Choice - 57 Questions - 12 of 25%**

(Office use only: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57)

1. A **Makefile** contains the following target: **dog: foo bar** which means:
  - † a. item dog depends on items foo and bar
  - b. items foo and bar depend on item dog
  - c. items dog and foo depend on item bar
  - d. item bar depends on items dog and foo
  - e. the syntax "dog:" is not valid in a Makefile target
2. A shell script named **dog** is executed as follows:
 

```
./dog "1 2" "a b c" z
```

 Inside the script is the line: **echo "\$2"**  
 What is the output on your screen from this line?
  - † a. a b c
  - b. 2
  - c. 2"
  - d. \$2
  - e. 1 2
3. Given my directory **dir** and my file **dir/foo** owned by me, which permissions allow me to delete the file from the directory, but not change the content (data) in the file?
  - † a. Permissions **300** on directory **dir** and **500** on file **dir/foo**
  - b. Permissions **100** on directory **dir** and **200** on file **dir/foo**
  - c. Permissions **100** on directory **dir** and **100** on file **dir/foo**
  - d. Permissions **300** on directory **dir** and **300** on file **dir/foo**
  - e. Permissions **500** on directory **dir** and **400** on file **dir/foo**
4. Given my directory **dir** and my file **dir/bar** owned by me, which permissions allow me to access and change the content (data) in the file **dir/bar** but not delete the file?
  - † a. Permissions **100** on directory **dir** and **200** on file **dir/bar**.
  - b. Permissions **500** on directory **dir** and **500** on file **dir/bar**.
  - c. Permissions **600** on directory **dir** and **200** on file **dir/bar**.
  - d. Permissions **300** on directory **dir** and **500** on file **dir/bar**.
  - e. Permissions **700** on directory **dir** and **300** on file **dir/bar**.

5. In response to the following command line: **read one two three** which user keyboard input line below will assign the text **bb** to the shell variable named **two**?
  - † a. **aa bb cc**
  - b. **one=aa two=bb three=cc**
  - c. **aa,bb,cc**
  - d. **aa:bb:cc**
  - e. **aa;bb;cc**
6. 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 of this shell command sequence:
 

```
PATH=/usr:/usr/bin:/etc:/bin ; foo
```

  - † a. **dad**
  - b. **mom**
  - c. **mom** followed by **dad**
  - d. **dad** followed by **mom**
  - e. **bash: foo: command not found**
7. 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 of this shell command sequence:
 

```
PATH=/bin/foo:/dev:/usr/bin/foo:/usr ; foo
```

  - † a. **bash: foo: command not found**
  - b. **mom**
  - c. **dad**
  - d. **mom** followed by **dad**
  - e. **dad** followed by **mom**
8. If **bat=55** and **cat=66** then which of the following command lines outputs only the date (and nothing else)?
  - † a. **[ bat = bat ] && date**
  - b. **[ bat -ne cat ] && date**
  - c. **[bat != cat] && date**
  - d. **[bat != cat] || date**
  - e. **[bat -eq 66] || date**
9. If **cow=cow** and **pig=pig** then which of the following command lines outputs only the date (and nothing else)?
  - † a. **test cow = cow && date**
  - b. **test cow -ne pig && date**
  - c. **[!cow = pig] && date**
  - d. **[cow -ne pig] || date**
  - e. **[cow!=cow] || date**
10. If **cow=cow** and **pig=pig** then which of the following command lines outputs only the date (and nothing else)?
  - † a. **[ cow = pig ] || date**
  - b. **[ cow -ne pig ] && date**
  - c. **test !cow = pig && date**
  - d. **[cow -eq pig] || date**
  - e. **test cow=cow || date**

11. If **dog** is an executable script containing the line: **umask 0002**  
what is the output on your screen of the following sequence of commands:  
**umask 0077 ; ./dog ; umask**
- † a. 0077  
b. 0002  
c. 0079  
d. 0075  
e. no output on screen
12. If **foo** is a file containing the first column of the output of the **last** command, which command line shows the most frequent login?
- † a. **sort <foo | uniq -c | sort -nr | head -1**  
b. **cat sort foo | uniq -c | sort -nr | head -1**  
c. **uniq -c foo | sort -nr | head -1**  
d. **sort | uniq -c | sort -nr | head -1 foo**  
e. **sort foo > uniq -c ; sort -nr uniq | head -1**
13. If **foo** is an executable script containing the line:  
**PATH=/bin ; export PATH**  
what is the output on your screen of the following sequence of commands:  
**PATH=/etc ; ./foo ; echo "\$PATH"**
- † a. /etc  
b. /bin  
c. foo  
d. /etc:/bin  
e. \$PATH
14. If **foo** is an executable script containing the line:  
**PATH=/bin ; export PATH**  
what is the output on your screen of the following sequence of commands:  
**PATH=/etc ; source ./foo ; echo "\$PATH"**
- † a. /bin  
b. /etc  
c. foo  
d. /etc:/bin  
e. \$PATH
15. If **a=1** and **b=1**, which command sequence correctly compares the two numbers as equal and prints the date?
- † a. **if [ \$a -eq \$b ] ; then date ; fi**  
b. **if test a -eq b ; then date ; fi**  
c. **if [ a = b ] ; then date ; fi**  
d. **if ( a == b ) ; then date ; fi**  
e. **if [ \$a=\$b ] ; then date ; fi**

16. If **a=aaa** and **b=bbb** then what is the output on your screen of the following command sequence: **if \$a = \$b ; then echo \$a ; fi**
- † a. **bash: aaa: command not found**  
b. **test: aaa: integer expression expected**  
c. **test: \$a: integer expression expected**  
d. **aaa**  
e. no output on screen
17. If **a=ant** and **b=bat** then what is the output on your screen of the following command sequence: **[ \$a = ant -a \$b = ant ] ; echo \$?**
- † a. 1  
b. 0  
c. the number 1 or 0 followed by another 1 or 0 on a new line  
d. **test: \$a: integer expression expected**  
e. no output
18. If **a=ant** and **b=bat** then what is the output on your screen of the following command sequence: **[ \$a = bat -o \$b = bat ] ; echo \$?**
- † a. 0  
b. 1  
c. the number 1 or 0 followed by another 1 or 0 on a new line  
d. **test: \$a: integer expression expected**  
e. no output
19. If a shell script named **foo** contains the line:  
**if [ "\$2" = '\$3' ] ; then echo SAME ; fi**  
then which of the following command lines will always produce **SAME** as output?
- † a. **./foo 2 '\$3' 1**  
b. **./foo \$1 '\$2' \$3**  
c. **./foo '\$1' "\$3" \$2**  
d. **./foo \$1 \$2 \$3**  
e. **./foo \$3 "\$2" \$1**
20. If the file **bat** contained the word **foo**, what would be the output on your screen of this two command sequence:  
**PATH=/bin/cat:/bin/who:/bin/ls ; cat bat**
- † a. **bash: cat: command not found**  
b. **bat**  
c. **foo**  
d. **cat: bat: No such file or directory**  
e. no output on screen
21. If the file **dog** contained the word **bar**, what would be the output on your screen of this two command sequence:  
**PATH=/etc/passwd:/bin/ls:/bin/who ; /bin/cat dog**
- † a. **bar**  
b. **dog**  
c. **/bin/cat: dog: No such file or directory**  
d. **bash: /bin/cat: command not found**  
e. no output on screen

22. What is the output on your screen of this two command sequence:  
**PATH=/bin/cat:/bin/sh:/bin/ls ; ls nosuchfile**
- † a. **bash: ls: command not found**
  - b. **bash: /bin/ls: command not found**
  - c. **ls: /bin/ls: command not found**
  - d. **ls: nosuchfile: No such file or directory**
  - e. **bash: /bin/sh: No such file or directory**
23. If variable **mt** might contain nothing (a null value - defined but empty), which command sequence correctly tests for this and prints the date?
- † a. **if [ "\$mt" = "" ] ; then date ; fi**
  - b. **if [ "\$mt" -z ] ; then date ; fi**
  - c. **if [ \$mt -eq "" ] ; then date ; fi**
  - d. **if [ '\$mt' = '' ] ; then date ; fi**
  - e. **if [ "\$mt" = \* ] ; then date ; fi**
24. In a directory containing one file named **dog**, what appears on your screen after this command line? **1>/dev/null ls \***
- † a. no output on screen
  - b. **\***
  - c. **dog**
  - d. **ls: \*: No such file or directory**
  - e. **bash: 1>/dev/null: command not found**
25. In a directory containing one file named **dog**, what appears on your screen after this command line? **2>/dev/null ls nosuchfile**
- † a. no output on screen
  - b. **nosuchfile**
  - c. **dog**
  - d. **ls: nosuchfile: No such file or directory**
  - e. **bash: 2>/dev/null: command not found**
26. In an empty directory, how many names are in file **bar** after this command line:  
**ls . nosuchfile 1>bar**
- † a. **1**
  - b. **2**
  - c. **3**
  - d. **4**
  - e. empty file (no data)
27. In an empty directory, how many names are in file **foo** after this command line:  
**ls nosuchfile . .. 2>foo**
- † a. **1**
  - b. **2**
  - c. **3**
  - d. **4**
  - e. empty file (no data)

28. In an empty directory, what appears on your screen after this command line?  
**ls 1>/dev/null nosuchfile**
- † a. **ls: nosuchfile: No such file or directory**
  - b. no output
  - c. **nosuchfile**
  - d. **ls: /dev/null: No such file or directory**
  - e. **ls: 1>/dev/null nosuchfile: No such file or directory**
29. In an empty directory, what appears on your screen after this command line?  
**ls 2>/dev/null nosuchfile**
- † a. no output
  - b. **nosuchfile**
  - c. **ls: nosuchfile: No such file or directory**
  - d. **ls: /dev/null: No such file or directory**
  - e. **ls: 2>/dev/null nosuchfile: No such file or directory**
30. In an empty directory, what is the output on your screen of these commands:  
**touch uu .u uv .v uw ; a="\*u \*v" ; echo "\$a"**
- † a. **\*u \*v**
  - b. **u\* v\***
  - c. **uu uv**
  - d. **uu .u uv .v**
  - e. **\$a**
31. In an empty directory, what is the length of the longest file name (including extension) after this sequence of commands?  
**date >four ; cp four five5 ; mv five5 hi ; bzip2 hi**
- † a. **6**
  - b. **5**
  - c. **4**
  - d. **3**
  - e. **7**
32. In an empty directory, what is the length of the longest file name created by the following two-command sequence: **a="1234 123 12 1" ; touch '\$a'**
- † a. **2 characters**
  - b. **3 characters**
  - c. **4 characters**
  - d. **1 character**
  - e. **13 characters**
33. Select the correct **bash** shell order of command line processing:
- † a. quotes, redirection, variables, GLOBs
  - b. quotes, variables, redirection, GLOBs
  - c. quotes, variables, GLOBs, redirection
  - d. quotes, GLOBs, variables, redirection
  - e. redirection, quotes, GLOBs, variables

34. The correct **g++** compiler suffix for a C++ source file is:
- † a. **.cpp**
  - b. **.cplus**
  - c. **.g++**
  - d. **.gpp**
  - e. **.C++**
35. The correct option to enable warning messages from the **g++** compiler is:
- † a. **-Wall**
  - b. **-wall**
  - c. **-warn**
  - d. **+Warn**
  - e. **-w**
36. What is the correct syntax to redirect both standard output and standard error into the same output file?
- † a. **ls -l >foo 2>&1**
  - b. **ls -l 2>&1 >foo**
  - c. **ls -l 1>foo 2>foo**
  - d. **ls -l 2&1>foo**
  - e. **ls -l 1&2>foo**
37. What is the output on your screen if a user signals an end-of-file from the keyboard during this command sequence? **read input || echo "\$?"**
- † a. **1**
  - b. no output on screen
  - c. **\$?**
  - d. **0**
  - e. an error message
38. What is the output on your screen of the following command sequence:
- ```
a=1 ; b=2 ; [ ! $a -ge $b ] ; echo $?
```
- † a. **0**
  - b. **1**
  - c. the number 1 or 0 followed by another 1 or 0 on a new line
  - d. **test: \$a: integer expression expected**
  - e. no output
39. What is the output on your screen of the following command sequence:
- ```
f=1 ; touch f ; [ ! -z $f ] ; echo $?
```
- † a. **0**
  - b. **1**
  - c. the number 1 or 0 followed by another 1 or 0 on a new line
  - d. **test: \$f: integer expression expected**
  - e. no output

40. What is the output on your screen of this sequence of three shell commands:
- ```
umask 162 ; touch newfile ; ls -l newfile
```
- † a. **-rw----r-- 1 foo foo 0 Apr 1 1:12 newfile**
  - b. **---xrw--w- 1 foo foo 0 Apr 1 1:12 newfile**
  - c. **----rw--w- 1 foo foo 0 Apr 1 1:12 newfile**
  - d. **-rw---xr-x 1 foo foo 0 Apr 1 1:12 newfile**
  - e. **-rw---x-w- 1 foo foo 0 Apr 1 1:12 newfile**
41. What is the output on your screen of this sequence of three shell commands:
- ```
umask 457 ; mkdir dir ; ls -ld dir
```
- † a. **d-wx-w---- 2 foo foo 64 Mar 9 1:12 dir**
  - b. **d-w--w---- 2 foo foo 64 Mar 9 1:12 dir**
  - c. **d-wx-w-rwx 2 foo foo 64 Mar 9 1:12 dir**
  - d. **dr--r-xrwx 2 foo foo 64 Mar 9 1:12 dir**
  - e. **dr-xr-xrwx 2 foo foo 64 Mar 9 1:12 dir**
42. What minimal permissions must you have on a directory to be able to execute successfully the command **ls .** from *inside* the directory?
- † a. **r-x**
  - b. **--x**
  - c. **r--**
  - d. **-wx**
  - e. **rw-**
43. Which command counts the number of Unix permission groups you are in?
- † a. **groups | wc**
  - b. **umask | wc**
  - c. **id | wc**
  - d. **echo /etc/groups | wc**
  - e. **wc /etc/groups**
44. Which command line below allows programs in the current directory to execute without preceding the names with **./**?
- † a. **PATH=/usr/bin:./bin**
  - b. **PATH=/usr/bin/.\$HOME**
  - c. **PATH=./\$HOME:/usr/bin**
  - d. **\$PATH=/usr/bin:./bin**
  - e. **\$PATH=./\$HOME:/usr/bin**
45. Which command line copies all the files from directory **a** to directory **b**?
- † a. **cd a ; tar czf /tmp/i . ; cd ../b ; tar xzf /tmp/i**
  - b. **cd a ; tar czf /tmp/i . ; cd ../b ; tar xvf /tmp/i**
  - c. **cd a ; tar xf /tmp/i . ; cd ../b ; tar czvf /tmp/i**
  - d. **cd a ; tar -r /tmp/i . ; cd ../b ; tar -rvx /tmp/i**
  - e. **cd a ; tar -rc /tmp/i . ; cd ../b ; tar -rx /tmp/i**

46. Which command line locates scripts in the `/bin` directory?
- † a. `file /bin/* | grep script`
  - b. `file /bin | grep script`
  - c. `cat /bin/* | file | grep script`
  - d. `cat /bin | file | grep script`
  - e. `ls /bin/* | file | grep script`
47. Which command line shows the file in `/bin` with the largest checksum?
- † a. `sum /bin/* | sort -nr | head -1`
  - b. `sum /bin | sort -nr | head -1`
  - c. `cat /bin/* | sum | sort -nr | head -1`
  - d. `cat /bin | sum | sort -nr | head -1`
  - e. `ls /bin/* | sum | sort -nr | head -1`
48. Which command line tells you the recursive count of all pathnames under the current directory and all subdirectories?
- † a. `find | wc`
  - b. `ls | wc`
  - c. `wc *`
  - d. `wc .`
  - e. `wc "$PATH"`
49. Which command sequence below always outputs just the date only if the first argument is either a file or a directory?
- † a. `if [ -f "$1" -o -d "$1" ]; then date ; fi`
  - b. `if [ "-f $1" || "-d $1" ]; then date ; fi`
  - c. `if [ "$1" -eq -f -o "$1" -eq -d ]; then date ; fi`
  - d. `if [ -f -o -d "$1" ]; then date ; fi`
  - e. `if [ -f || -d "$1" ]; then date ; fi`
50. Which command sequence correctly searches for `foo` and then prints the date if it is found inside the file `bar`?
- † a. `if grep <bar foo ; then date ; fi`
  - b. `if [ grep foo bar ] ; then date ; fi`
  - c. `if test foo bar ; then date ; fi`
  - d. `if [ grep foo <bar ] ; then date ; fi`
  - e. `if test -r foo bar ; then date ; fi`
51. Which command tells you the full absolute pathname of the `lynx` command?
- † a. `whereis lynx`
  - b. `absolute lynx`
  - c. `whereis | grep lynx`
  - d. `echo "$PATH" | grep lynx`
  - e. `absolute "$PATH" | grep lynx`

52. Which line below passes three *separate* arguments to the `cat` command when placed inside a shell script named `foo` invoked by the command line:
- ```
./foo one two three
```
- † a. `cat "$@"`
  - b. `cat "$*"`
  - c. `cat "$#"`
  - d. `cat "$1 $2 $3"`
  - e. `cat "$? $? $?"`
53. Which line below puts the count of the number of lines in the password file into the variable `foo`?
- † a. `foo=$( wc -l </etc/passwd )`
  - b. `foo=$( wc -lines /etc/passwd )`
  - c. `foo=$( wc /etc/passwd | awk "echo $1" )`
  - d. `foo=$( wc -l /etc/passwd | awk "print $1" )`
  - e. `foo=$( awk -F: /etc/passwd | wc -l )`
54. Which of these commands makes a file owned by me, also executable by me (without changing any other permissions)?
- † a. `chmod u+x myfile`
  - b. `chmod x+u myfile`
  - c. `chmod x=u myfile`
  - d. `umask 100 myfile`
  - e. `chmod 100 myfile`
55. What is the meaning of this SysV init script pathname: `/etc/rc3.d/S04httpd`
- † a. start the web server in run level 3, sort order 4
  - b. start the web server in run level 4, sort order 3
  - c. stop the web server in run level 3, sort order 4
  - d. start the web server in run level 04; stop it in run level 3
  - e. start the web server in run level 3; stop it in run level 04
56. Which of these lines fetches an entire raw index web page using the `http` protocol?
- † a. `GET /index.html HTTP/1.0`
  - b. `HEAD /index.html HTTP/1.0`
  - c. `HEAD HTTP/1.0 index.html`
  - d. `GET HTTP/1.0 index.html`
  - e. `HEAD HTTP/1.0 /index.html`
57. Which line will show you the default run level for your system?
- † a. `grep initdefault /etc/inittab`
  - b. `grep default /etc/rc?.d`
  - c. `grep init /etc/rc?.d/*`
  - d. `grep default /etc/init.d`
  - e. `grep init /etc/init.d/*`

**Answer Key - NET 2003 – Ian Allen – Winter 2005 - NET 2003 Unix Test  
#2 - 25%**

Office use only: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45  
46 47 48 49 50 51 52 53 54 55 56 57

```

1. a          45. a
2. a          46. a
3. a          47. a
4. a          48. a
5. a          49. a
6. a          50. a
7. a          51. a
8. a          52. a
9. a          53. a
10. a         54. a
11. a         55. a
12. a         56. a
13. a         57. a
14. a
15. a
16. a
17. a
18. a
19. a
20. a
21. a
22. a
23. a
24. a
25. a
26. a
27. a
28. a
29. a
30. a
31. a
32. a
33. a
34. a
35. a
36. a
37. a
38. a
39. a
40. a
41. a
42. a
43. a
44. a

```

Count of a: 57 100%

With 5 choices: 57

```

1 2 3 4 5 6 7 8 9 10 11 12
13 14 15 16 17 18 19 20 21
22 23 24 25 26 27 28 29 30
31 32 33 34 35 36 37 38 39
40 41 42 43 44 45 46 47 48
49 50 51 52 53 54 55 56 57

```

```

Macro .cmd split no indent: 0
Macro .cmd split with indent: 23
Macro .ans splits: 0

```