UNIX Reference Sheets

CSE 2031

Fall 2010

Wildcards (File Name Substitution)

- `?` match single character
- `*` match any number of characters
- `[…]` match any character in the list enclosed by `[ ]`
- We can combine different wildcards.

File Manipulation Commands

```
ls, cp, mv, rm
touch
pwd, mkdir, rmdir
cd
chmod, chown, chgrp
find
% find . -name "e*.c"
% find ~ -type f
```

Commonly Used Commands

- Get on-line help with **man**
- Some commonly used commands

```
cat, more
who, echo, date
cmp, diff
```

```
sort
wc
ps, kill
history
grep
-i case insensitive
-l display only file name
-n display line numbers
-v lines that do not
   contain pattern
```

Commonly Used Commands (2)

```
% wc file
% wc –c file
% wc –w file
% wc –l file
```

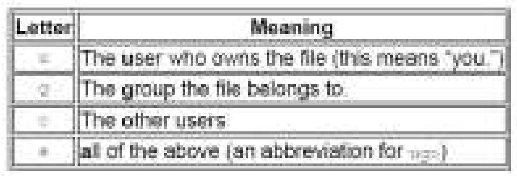
`sort –r`
reverse normal order

`sort –n`
numeric order

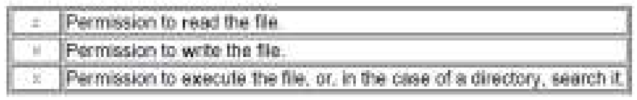
`sort –nr`
reverse numeric order

`sort –f`
case insensitive

File/Directory Permissions



Letter	Meaning
u	The user who owns the file (this means "you.")
g	The group the file belongs to.
o	The other users
a	all of the above (an abbreviation for ugo)



r	Permission to read the file.
w	Permission to write the file.
x	Permission to execute the file, or, in the case of a directory, search it.

Pre-defined "Variables"

- **$#** represents the number of command line arguments
- **$*** represents all the command line arguments
- **$@** represents all the command line arguments
- **$$** represents the process ID of the shell
- **$?** represents the exit status code of the command last executed

User Variables

```
name=value
read name
echo $name
```

- **expr utility**

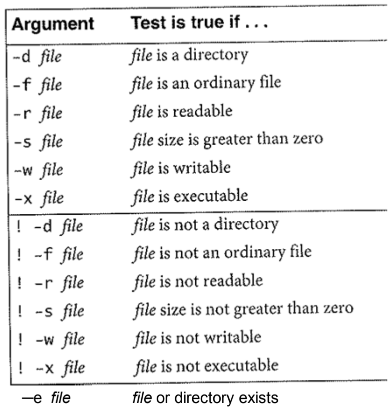
```
  sum=`expr $op1 + $op2`
  echo $sum
```

if Statement and test Command

```
if condition
then
    command(s)
elif condition_2
then
    command(s)
else
    command(s)
fi
```

9

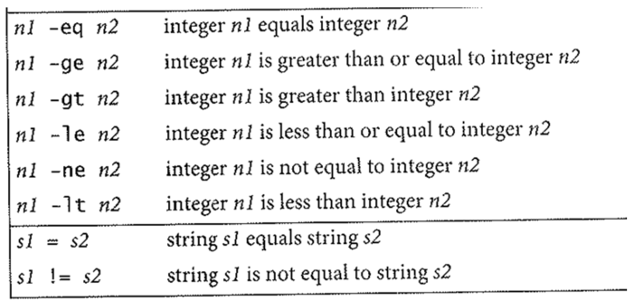
test Command



Argument	Test is true if . . .
-d *file*	*file* is a directory
-f *file*	*file* is an ordinary file
-r *file*	*file* is readable
-s *file*	*file* size is greater than zero
-w *file*	*file* is writable
-x *file*	*file* is executable
! -d *file*	*file* is not a directory
! -f *file*	*file* is not an ordinary file
! -r *file*	*file* is not readable
! -s *file*	*file* size is not greater than zero
! -w *file*	*file* is not writable
! -x *file*	*file* is not executable
–e *file*	*file* or directory exists

10

test Command (2)



n1 -eq *n2*	integer *n1* equals integer *n2*
n1 -ge *n2*	integer *n1* is greater than or equal to integer *n2*
n1 -gt *n2*	integer *n1* is greater than integer *n2*
n1 -le *n2*	integer *n1* is less than or equal to integer *n2*
n1 -ne *n2*	integer *n1* is not equal to integer *n2*
n1 -lt *n2*	integer *n1* is less than integer *n2*
s1 = *s2*	string *s1* equals string *s2*
s1 != *s2*	string *s1* is not equal to string *s2*

- Parentheses can be used for grouping test conditions.
- Logical operators: `!` `||` `&&`

11

for Loops

```
for variable in list
do
    command(s)
done
```

- ***variable*** is a user-defined variable.
- ***list*** is a sequence of strings separated by spaces.

12

while Loops

```
while condition
do
    command(s)
done
```

- Command **test** is often used in ***condition***.
- Execute ***command(s)*** when ***condition*** is met.

until Loops

```
until condition
do
    command(s)
done
```

- Command **test** is often used in ***condition***.
- Exit loop when ***condition*** is met.

case Statement

```
case variable in
pattern1) command(s);;
pattern2) command(s);;
. . .
patternN) command(s);;
*)        command(s);;  # all other cases
esac
```

Shell Functions

```
# function declaration and implementation
my_function()
{
   commands
}


# caller
my_function
```