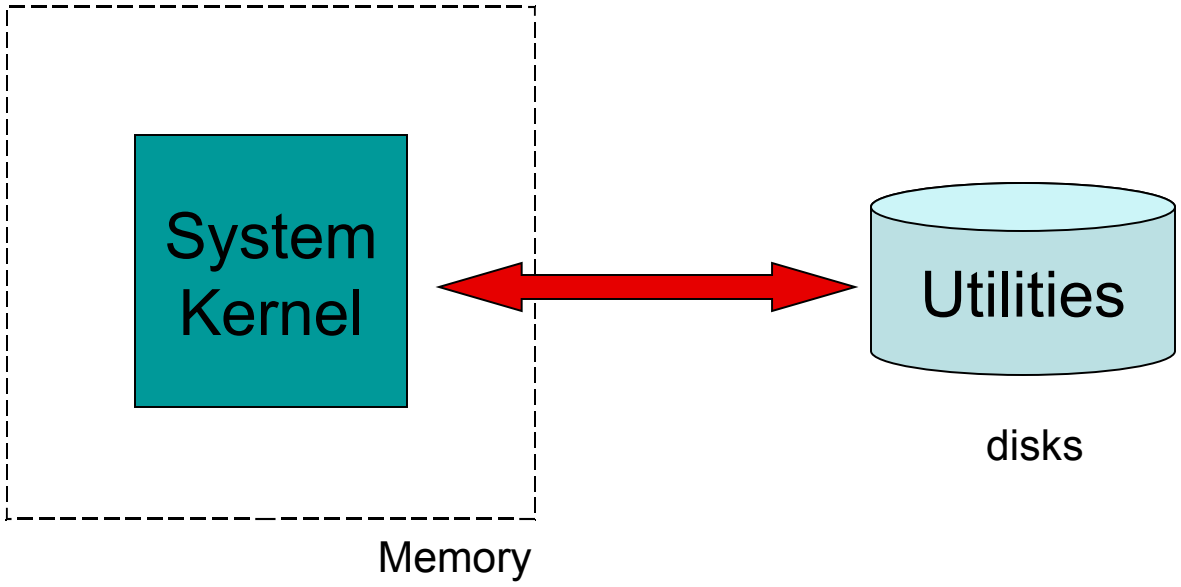
Introduction to Shell Scripting

Joey Bernard, Computational Research Consultant

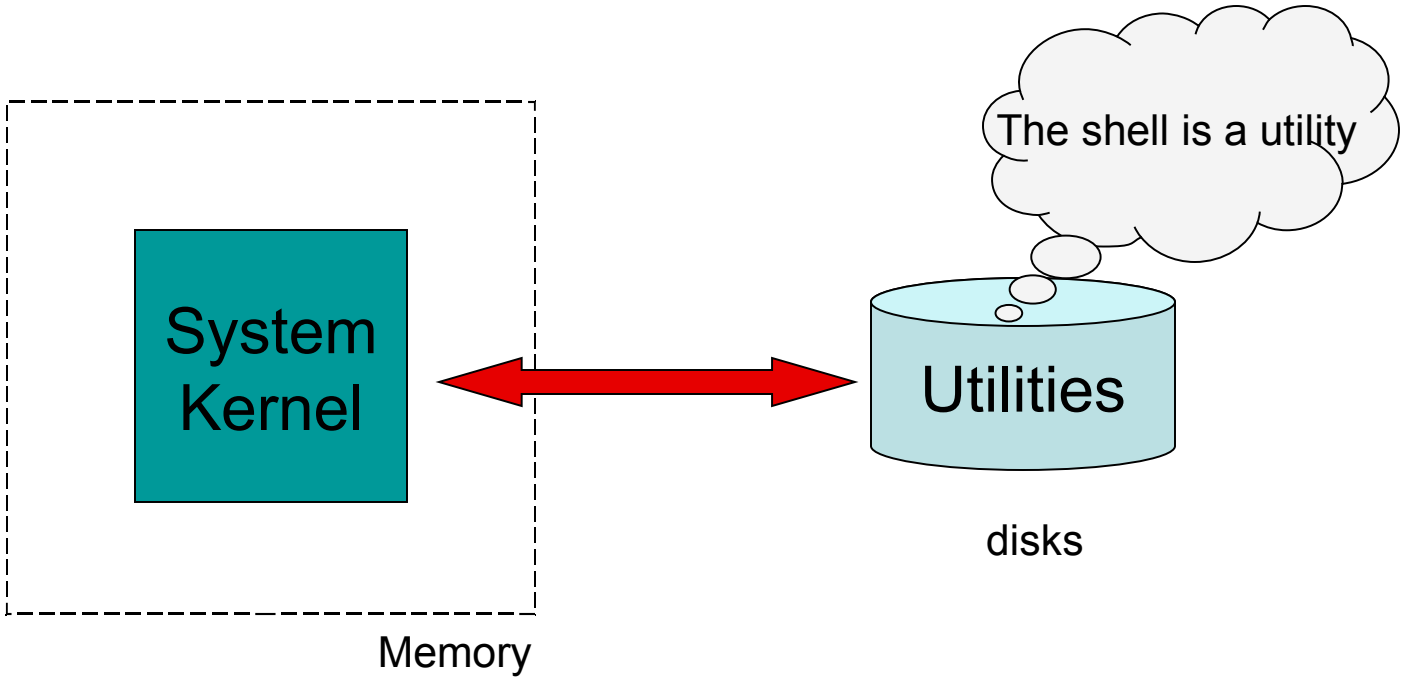
What is the Shell?

#The Unix/Linux sytem is divided into the *kernel* and *utilities*



What is the Shell?

#The Unix/Linux sytem is divided into the *kernel* and *utilities*



Running Shell Scripts

- Running directly
  - /bin/sh my_shell.sh
  - /bin/bash my_shell.sh
- Making them executable
  - #$/bin/sh
  - #$/bin/bash

Let’s write our first shell script

Using vi or emacs type the following into a file called my first_shell.sh

Let's write our first shell script

Using vi or emacs type the following into a file called my first_shell.sh

```
echo "The date and time today is" \ `date` \
echo
who
echo
echo "The number of login windows is" $(who | wc -l)
echo
echo "What is my user name:" $(who am i) "and where am i:" $(pwd)
```

Let's write our first shell script

Using vi or emacs type the following into a file called my first_shell.sh

Let's look closely at the commands and syntax

```
echo "The date and time today is" \ `date`
echo
who
echo
echo "The number of login windows is" $(who | wc -l)
echo
echo "What is my user name:" $(who am i) "and where am i:" $(pwd)
```

The output is:

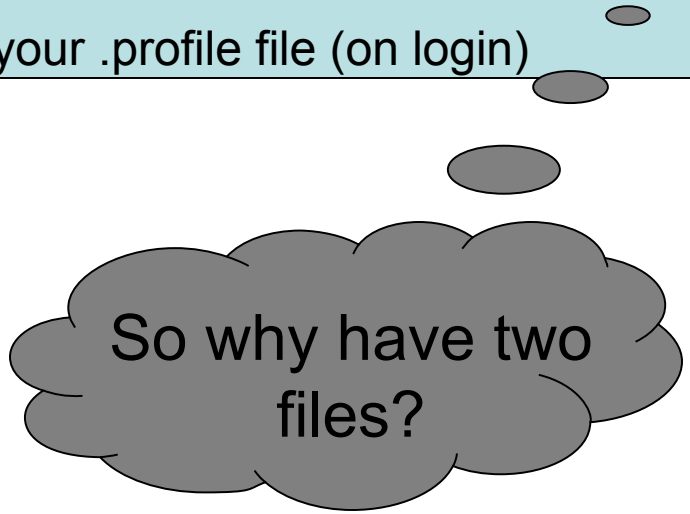
```
The date and time today is  Thu Oct 9 12:40:07 ADT 2008

srazul   pts/0        Oct  9 11:24 (141.109.52.35)
evorobyo pts/1        Oct  9 07:34 (76.11.66.169)
sgupta   pts/2        Oct  9 09:12 (puffin.bioc.mun.ca)
mcoates  pts/3        Sep 10 12:23 (134.153.141.136)
ghuang   pts/4        Oct  9 11:43 (141.109.36.40)
aelsherb pts/5        Sep 29 16:58 (doc.chem.mun.ca)
jzhu     pts/6        Oct  8 15:46 (coriolis.physics.mun.ca)
ishort   pts/7        Oct  8 09:12 (crux.smu.ca)
rkarsten pts/8        Sep 30 12:28 (mathfluids.acadiau.ca)
ghuang   pts/9        Oct  9 11:47 (141.109.36.40)
jchaffey pts/10       Oct  9 11:10 (0xtp.mar.dfo-mpo.gc.ca)
pyuet    pts/11       Oct  9 11:20 (landau-four-forty-six.mit.edu)
jbernard pts/12       Oct  9 11:25 (bernardmac.cs.unb.ca)
aelsherb pts/13       Oct  6 14:08 (doc.chem.mun.ca)
rdickson pts/14       Oct  9 11:57 (df01a.vpn.dal.ca)
yhuang   pts/16       Oct  9 12:02 (141.109.36.42)
jramsey  pts/23       Oct  8 11:51 (ttauri.smu.ca)
jzhu     pts/26       Oct  8 15:53 (coriolis.physics.mun.ca)

The number of login windows is 18

What is my user name: srazul pts/0 Oct 9 11:24 (141.109.52.35) and where am i: /home/srazul
```

Location of script

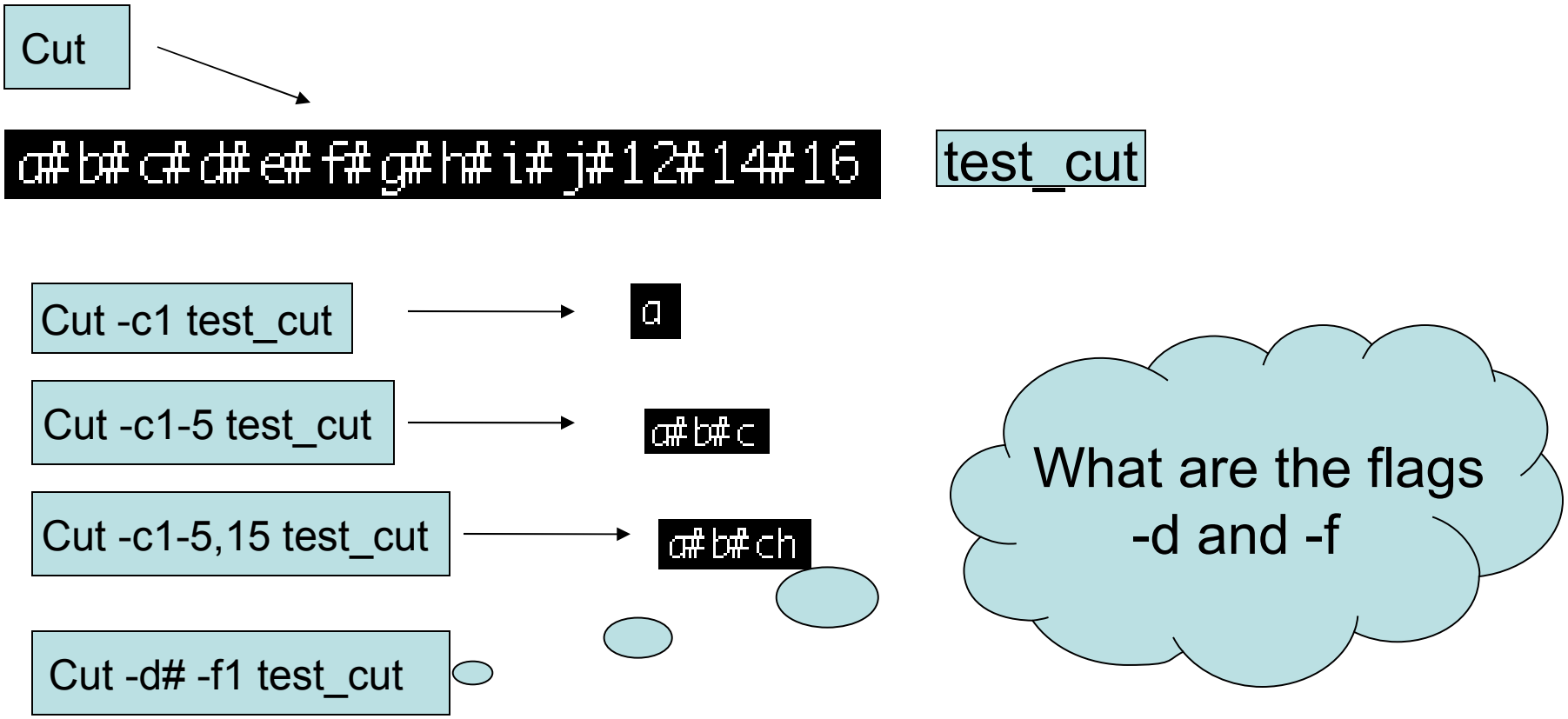
This script resides in your /home/<username>
To launch the script everytime you log in, you can

a) Put in into your .bashrc file or better yet

b) Put the contents in your .profile file (on login)



Let's look at some useful commands



Let's look at some useful commands

test_paste1

```
aaa
bbb
ccc
ddddd
eeeee
fffff
```

test_paste2

```
902 555 1234
902 555 4567
902 555 2365
902 555 5673
902 555 0989
902 555 6575
```

paste

paste test_paste1 test_paste2 > new

```
aaa     902 555 1234
bbb     902 555 4567
ccc     902 555 2365
ddddd   902 555 5673
eeeee   902 555 0989
fffff   902 555 6575
```

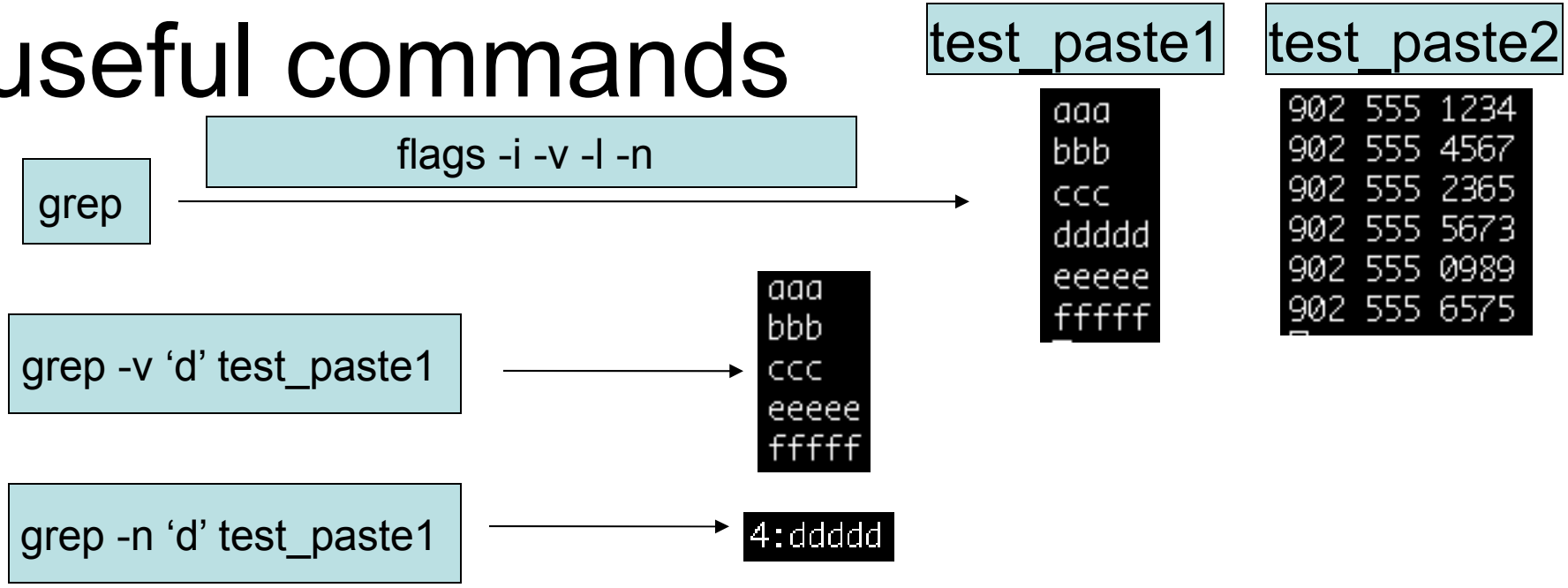
paste -d '%' test_paste1 test_paste2 > new

```
aaa%902 555 1234
bbb%902 555 4567
ccc%902 555 2365
ddddd%902 555 5673
eeeee%902 555 0989
fffff%902 555 6575
```

paste -s test_paste1 > new

```
aaa     bbb     ccc     ddddd   eeeee   fffff
```

Let's look at some useful commands



PATTERNS	MATCHED STRINGS
a.c	a[any single character]c, e.g. abc, a1c, a c.
a[xyz]c or a[x-z]c	axc, ayc, and azc only
a[^xyz]c	a[any single character but x, y, or z]c
ab*c	a[0 or more b]c, e.g. ac, abc, abbbc
ab+c	a[1 or more b]c, e.g. abc, abbbc
^abc	abc only at the beginning of a line
abc$	abc only at the end of a line
a(bc|de)f	abcf and adef

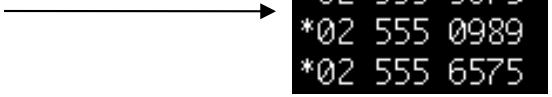
Let's look at some useful commands

sed

test_paste2

```
902 555 1234
902 555 4567
902 555 2365
902 555 5673
902 555 0989
902 555 6575
```

sed 's/9/0/' test_paste1



```
*02 555 1234
*02 555 4567
*02 555 2365
*02 555 5673
*02 555 0989
*02 555 6575
```

Let's look at some useful commands

sed

test_paste2

```
902 555 1234
902 555 4567
902 555 2365
902 555 5673
902 555 0989
902 555 6575
```

sed 's/9/0/g' test_paste2

```
*02 555 1234
*02 555 4567
*02 555 2365
*02 555 5673
*02 555 0*8*
*02 555 6575
```

sed '5d' test_paste2

```
902 555 1234
902 555 4567
902 555 2365
902 555 5673
902 555 6575
```

sed '/3/d' test_paste2

```
902 555 4567
902 555 0989
902 555 6575
```

sed -n '1,3p' test_paste2

Prints out first three lines

sed 's/^/hello /g' test_paste2
sed 's/$/ hello/g' test_paste2

```
hello 902 555 1234
hello 902 555 4567
hello 902 555 2365
hello 902 555 5673
hello 902 555 0989
hello 902 555 6575
```

Let's look at some useful commands

test_paste2

```
902 555 1234
902 555 4567
902 555 2365
902 555 5673
902 555 0989
902 555 6575
```

sed

sed 's/9/0/g' test_paste2

```
*02 555 1234
*02 555 4567
*02 555 2365
*02 555 5673
*02 555 0*8*
*02 555 6575
```

sed '5d' test_paste2

```
902 555 1234
902 555 4567
902 555 2365
902 555 5673
902 555 6575
```

sed '/3/d' test_paste2

```
902 555 4567
902 555 0989
902 555 6575
```

sed -n '1,3p' test_paste2

Prints out first three lines

sed 's/^/hello /g' test_paste2
sed 's/$/ hello/g' test_paste2

```
902 555 1234 hello
902 555 4567 hello
902 555 2365 hello
902 555 5673 hello
902 555 0989 hello
902 555 6575 hello
```

Let’s put these to work!

Suppose you have a fifty jobs on the cluster, some in *r* state, some in *qw* state and others in *Eqw* state

How do you delete your *Eqw* jobs efficiently?

Let's put these to work!

Suppose you have fifty jobs on the cluster, some in *r* state, some in *qw* state and others in *Eqw* state

How do you delete your *Eqw* jobs efficiently?

```
qstat -u <username> > temp1

grep 'Eqw' temp1 > temp2

cut -c1-8 temp2 > temp3

sed 's/^/qdel /g' temp3 > temp4

mv temp4 del.sh

sh del.sh
```

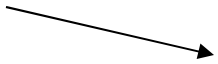
You could use this script!
Let's see what it does.

Eqw.sh

```
 44988 0.30025 refmod3K   evorobyo      r     10/06/2008 19:00:27 medium.q@cl047.smu.acenet.ca       4
 45112 0.25813 mBg_60A2_1 sgupta        r     10/09/2008 11:12:01 medium.q@cl049.smu.acenet.ca       8
 44650 0.25022 Li128_8_30 atewelde      r     09/28/2008 16:53:32 medium.q@cl050.smu.acenet.ca       8
 44644 0.25022 Li128_7_30 atewelde      r     09/27/2008 00:25:17 medium.q@cl052.smu.acenet.ca       8
 45107 0.25813 mBg_60A2_1 sgupta        r     10/07/2008 15:09:43 medium.q@cl053.smu.acenet.ca       8
```

Let’s look at more useful tools!

If
then
else
fi



```
user="$1"

if who | grep "^$user " > /dev/null
then
     echo "$user is logged ON"
else
     echo "$user is not logged on"
fi
```

test_if.sh

Let’s look at more useful tools!

If
then
else
fi

```
user="$1"

if who | grep "^$user " > /dev/null
then
     echo "$user is logged ON"
else
     echo "$user is not logged on"
fi
```

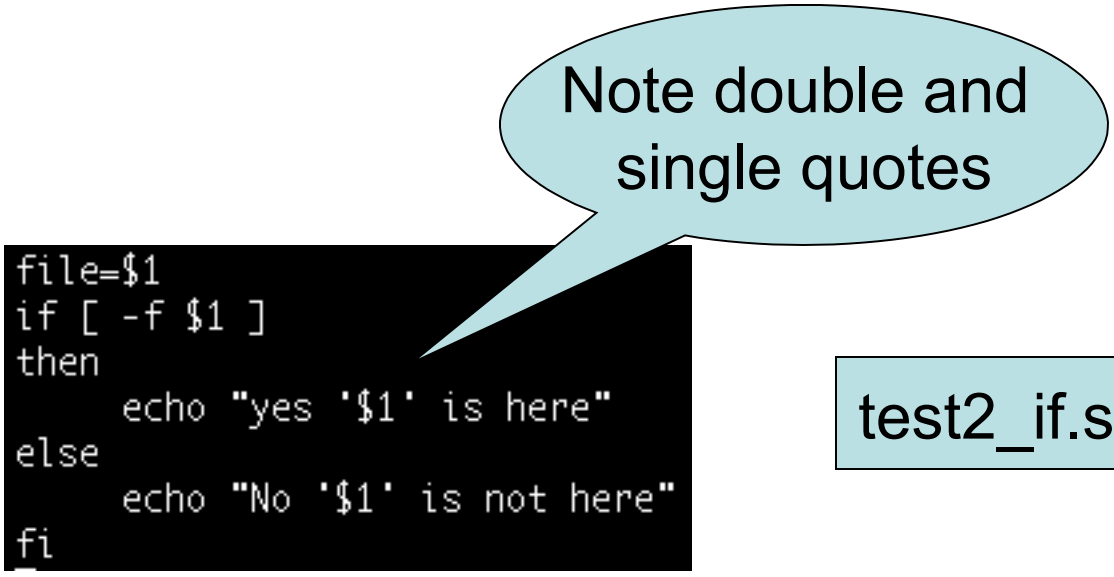
test_if.sh

Lets look at these!

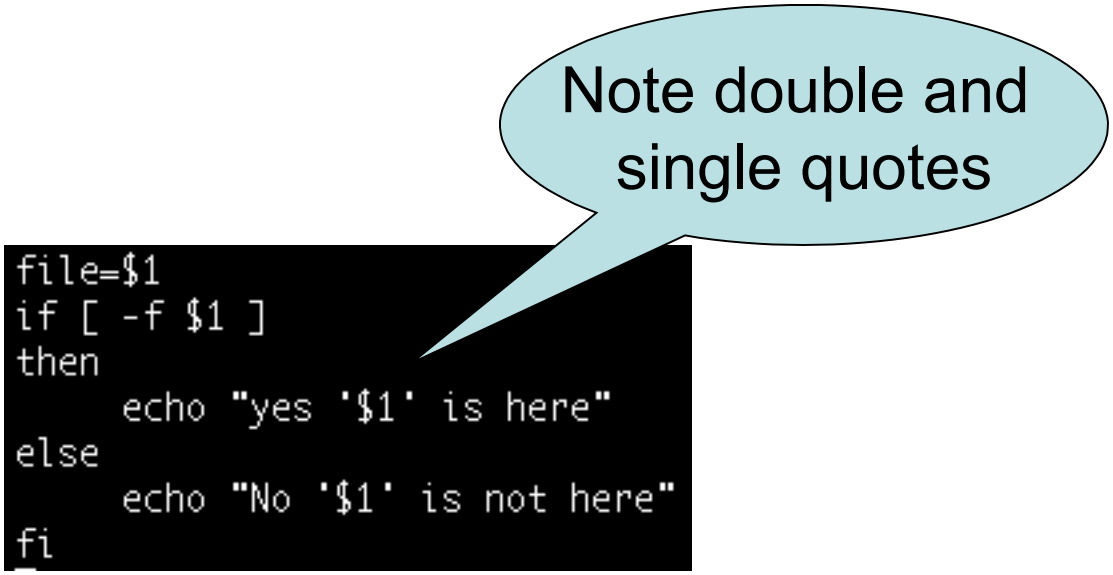
Let's look at some file testing tools!

Operator	Returns TRUE (exit status of 0) if
-d *file*	---------> *file is a directory*
-e *file*	---------> *file exits*
-f *file*	---------> *file is an ordinary file*
-r *file*	---------> *file is readable by the process*
-s *file*	---------> *file has nonzero length*
-w *file*	---------> *file is writable by the process*
-x *file*	---------> *file is executable*
-L *file*	---------> *file is a symbolic link*

Let’s look at an example



Let’s look at an example



Let’s look at another example

```
hour=$(date | cut -c12-13)
echo $hour
if [ "$hour" -ge 0 -a "$hour" -le 11 ]
then
       echo "Good Morning"
elif [ "$hour" -ge 12 -a "$hour" -le 17 ]
then
      echo "Good Afternoon"
else
      echo "Good evening"
fi
```

test3_if.sh

Let's look at another example

```
hour=$(date | cut -c12-13)
echo $hour
if [ "$hour" -ge 0 -a "$hour" -le 11 ]
then
       echo "Good Morning"
elif [ "$hour" -ge 12 -a "$hour" -le 17 ]
then
      echo "Good Afternoon"
else
      echo "Good evening"
fi
```

test3_if.sh

What other's are there? -gt -greater than
-lt -less than
-eq -equal to
-ne -not equal to

Let's look at another example

```
hour=$(date | cut -c12-13)
echo $hour
if [ "$hour" -ge 0 -a "$hour" -le 11 ]
then
       echo "Good Morning"
elif [ "$hour" -ge 12 -a "$hour" -le 17 ]
then
      echo "Good Afternoon"
else
      echo "Good evening"
fi
```

test3_if.sh

What other's are there?

-o Logical OR operator

Let's look at another example

```
user=$1
hour=$(date | cut -c12-13)
echo $hour
#
if [ "$hour" -ge 0 -a "$hour" -le 17 ] &&  [ "$user" =  srazul ]
then
       echo "Good Day"
else
      :
fi
```

test4_if.sh

We also have the && (AND) and the || (OR) operators for combining commands

Let’s look at another example

```
user=$1
hour=$(date | cut -c12-13)
echo $hour
#
if [ "$hour" -ge 0 -a "$hour" -le 17 ] &&  [ "$user" =  srazul ]
then
       echo "Good Day"
else
      :
fi
```

test4_if.sh

The null command

Let's look at another example

while
do
done

```
#
i=1
while [ "$i" -le 5 ]
do
      echo $i
      i=$((i+1))
done
#
```

test_do.sh

+	plus
-	minus
*	multiplication
/	division
%	modulus

Let's look at another example

while
do
done

```
#
i=1
while [ "$i" -le 5 ]
do
      echo $i
      i=$((i+1))
done
#
```

test_do.sh

+	plus
-	minus
*	multiplication
/	division
%	modulus

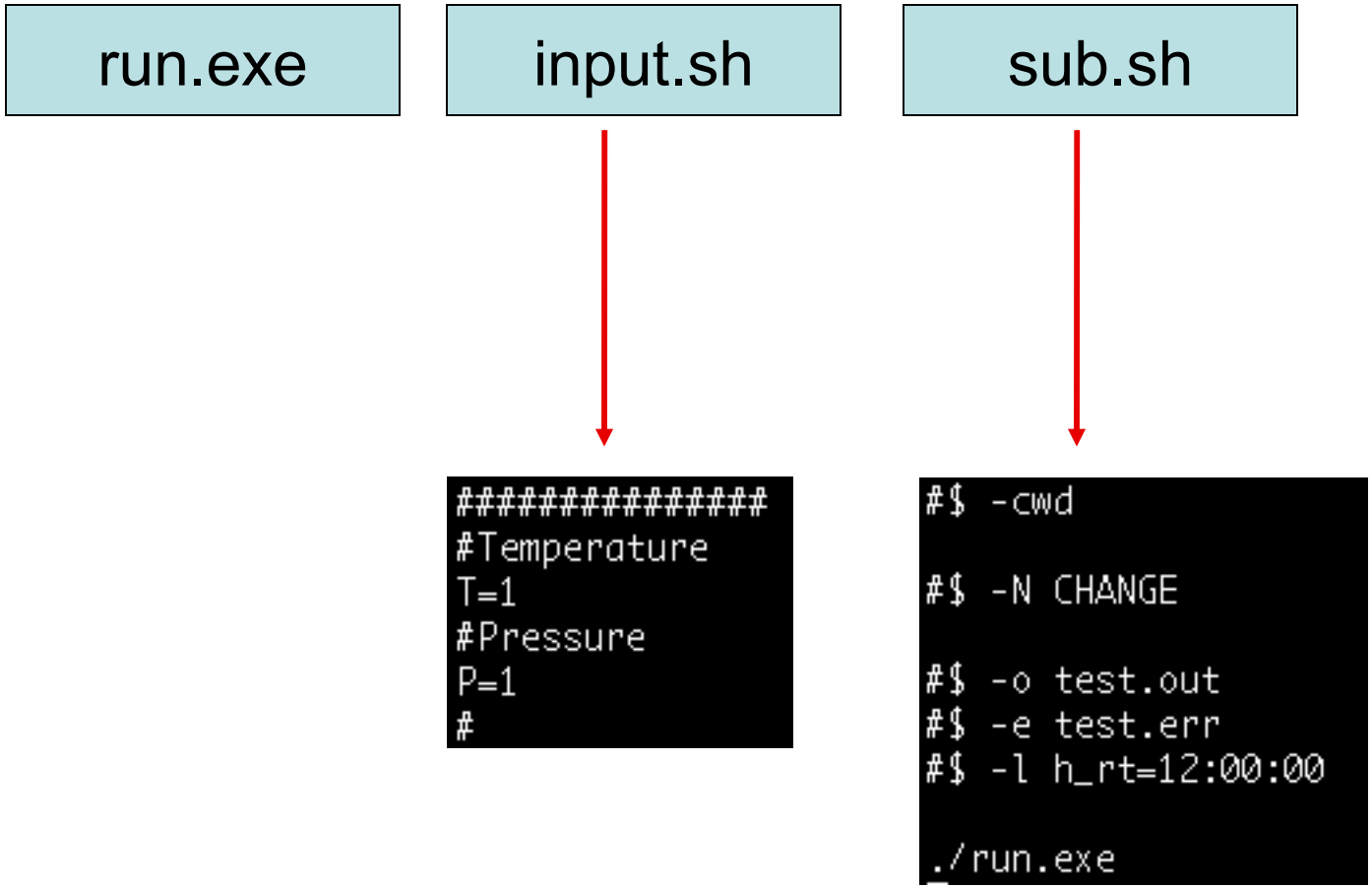
until
do
done

```
#
i=1
until [ "$i" -eq 6 ]
do
      echo $i
      i=$((i+1))
done
#
```

test2_do.sh

Let’s look at a useful script at work!

You log into your account:
you want to submit 20 jobs,
you first have to change the Temperature
on your input file and submit your job
with a unique job tag and submit script


run.exe
input.sh
sub.sh
################
#Temperature
T=1
#Pressure
P=1
#
#$ -cwd
#$ -N CHANGE
#$ -o test.out
#$ -e test.err
#$ -l h_rt=12:00:00
./run.exe

Let's look closely at this script

```
i=1
j=1
while [ "$i" -lt 20 ]
do
     echo $i
     i=$((i+1))
#
    mkdir mydir."$i"
#
    cp run.exe mydir."$i"
    cp input.sh mydir."$i"
    cp sub.sh sub"$i".sh
    mv sub"$i".sh mydir."$i"
#
    j=$((j+1))
#
    cd mydir."$i"
    sed 's/T=1/T='$j'/g' input.sh > temp
    mv temp input"$i".sh
    rm input.sh
#
    sed 's/CHANGE/NO.'$i'/g' sub"$i".sh > temp
    mv temp sub"$i".sh
    cd ..
#
done
#
```

mkdir.sh