CIS241

System-Level Programming and Utilities

bash Conditionals

Erik Fredericks, frederer@gvsu.edu Fall 2025

Based on material provided by Erin Carrier, Austin Ferguson, and Katherine Bowers

if

General form:

```
if CONDITION
then
# do the things
fi
```

or:

```
if CONDITION; then
# do the things
fi
```

CONDITION?

Think of your if tests - those are conditionals

```
if [ $1 -lt 10 ]; then
# do things here
fi
```

What does this do?

MORE

Operator	Description
! EXPRESSION	The EXPRESSION is false.
-n STRING	The length of STRING is greater than zero.
-z STRING	The lengh of STRING is zero (ie it is empty).
STRING1 = STRING2	STRING1 is equal to STRING2
STRING1 != STRING2	STRING1 is not equal to STRING2
INTEGER1 -eq INTEGER2	INTEGER1 is numerically equal to INTEGER2
INTEGER1 -gt INTEGER2	INTEGER1 is numerically greater than INTEGER2
INTEGER1 -It INTEGER2	INTEGER1 is numerically less than INTEGER2
-d FILE	FILE exists and is a directory.
-e FILE	FILE exists.
-r FILE	FILE exists and the read permission is granted.
-s FILE	FILE exists and it's size is greater than zero (ie. it is not empty).
-w FILE	FILE exists and the write permission is granted.
-x FILE	FILE exists and the execute permission is granted.

CIS241 | Fredericks | F25 | 22-bash-conditionals

Conditions

• Can also run man test

Alternative - arithmetic expression!

```
if (( $1 < 10 )); then
# do things
fi
```

And then, the else

```
if CONDITION; then
    # do thing 1
elif CONDITION2; then
    # do thing 2
else
    # do thing 3
fi
```

Boolean operators

```
if [ $1 -lt 10 ] && [ $1 -gt 0 ]; then
    # do stuff
fi

if [ -z $2 ] || [ $2 = "start" ]; then
    # do stuff
fi
```

And finally, the case

```
test1="HELLO"
case $test1 in
HELLO | HI)
  echo "case successful: $test1";;
ANOTHER)
  echo "a different case" ;;
"AND ANOTHER")
  echo "different yet again";;
* \
  echo "default case" ;;
esac
```

https://linuxize.com/post/bash-case-statement/