



Introduction to Bash video lecture

10 - Redirecting streams and
creating files from output



Creating files from command output

- In Bash, there are three streams for data flowing to and from commands
- Standard In(put) - default stream for data consumed by the command
- Standard Out(put) - Expected printouts from commands (like `ls` etc)
- Standard Err(or) - Dedicated for error messages
- All of the above can be *redirected* to (and for Std In, from) files

Interactive commands, line-based commands

- Many commands for text processing can be run interactively
- Bash is an example
- These commands are line-based, they operate on lines
- That's why you need to press Enter to issue a command

Running cat interactively

- cat reads lines from standard in and prints each line out
- It can read and print lines from files, or from standard in

```
rikard@newdelli:~/tempo$ cat
one
one      <- output from cat
two
two      <- output from cat
three
three    <- output from cat
rikard@newdelli:~/tempo$
```

Running grep interactively

```
rikard@newdelli:~/tempo$ grep bingo
```

```
lingo
```

```
dingo
```

```
zingo
```

```
bingo
```

```
bingo          <- output from grep
```

```
rikard@newdelli:~/tempo$
```

Redirecting standard in

- You can use Bash to redirect standard in to come from a file instead of the keyboard

```
rikard@newdelli:~/tempo$ cat < meals.txt
Breakfast: Egg and tea
Lunch: Fish and chips
Snack: Sandwich and juice
Dinner: Stake and sallad
Breakfast: Egg and coffee
Lunch: Hamburger and coke
Snack: Peanuts and beer
Dinner: Pizza
Breakfast: Sandwich and milk
Lunch: Fish and potato
Snack: Apple
Dinner: Pasta and wine
rikard@newdelli:~/tempo$
```

Redirecting standard in

- You can use Bash to redirect standard in to come from a file instead of the keyboard

```
rikard@newdelli:~/tempo$ grep ch < meals.txt
Lunch: Fish and chips
Snack: Sandwich and juice
Lunch: Hamburger and coke
Breakfast: Sandwich and milk
Lunch: Fish and potato
rikard@newdelli:~/tempo$
```

Redirecting standard out

- You can use Bash to redirect standard out to go to a file, rather than the terminal
- Doing so will create or overwrite the file

```
rikard@newdelli:~/tempo$ grep Lunch meals.txt > lunches.txt
```

```
rikard@newdelli:~/tempo$ cat lunches.txt
```

```
Lunch: Fish and chips
```

```
Lunch: Hamburger and coke
```

```
Lunch: Fish and potato
```

```
rikard@newdelli:~/tempo$
```


Copying text from the terminal

- Two clipboards
- Things you highlight with the mouse is copied and can be pasted via the middle button
- You can also highlight and copy with Ctrl-Ins and paste with Shift-Ins
- If you are used to Ctrl-C and Ctrl-V, you need to reprogram your brain
- Ctrl-C is used to make the terminal send the Interrupt (Terminate) signal to the program in the foreground

```
rikard@newdelli:~/bash-intro/text-files$ cat  
^C  
rikard@newdelli:~/bash-intro/text-files$
```

Redirecting standard out with append

- Use >> to redirect and append (first time the file will be created)

```
rikard@newdelli:~/tempo $ grep Lunch meals.txt > lunches_and_dinners.txt
```

```
rikard@newdelli:~/tempo $ grep Dinner meals.txt >> lunches_and_dinners.txt
```

```
rikard@newdelli:~/tempo $ cat lunches_and_dinners.txt
```

```
Lunch: Fish and chips
```

```
Lunch: Hamburger and coke
```

```
Lunch: Fish and potato
```

```
Dinner: Stake and sallad
```

```
Dinner: Pizza
```

```
Dinner: Pasta and wine
```

Error messages go to Standard err

```
rikard@newdelli:~/tempo$ grep Breakfast meels.txt > breakfasts.txt
grep: meels.txt: No such file or directory
rikard@newdelli:~/tempo$ ls
breakfasts.txt  lunches_and_dinners.txt  lunches.txt  meals.txt
rikard@newdelli:~/tempo$
```

```
# What is inside breakfasts.txt ?
# The file to redirect to is created as an empty file!
# It's a good thing that error messages aren't redirected
# unless we explicitly ask for it...
```

Redirecting standard err and standard out

```
$ ls pictures/ 2>> errors.txt >> picture-and-movies.txt
$ ls movies/ 2>> errors.txt >> picture-and-movies.txt
$ ls images/ 2>> errors.txt >> picture-and-movies.txt
$ ls films/ 2>> errors.txt >> picture-and-movies.txt
```

```
$ cat errors.txt
ls: cannot access 'images/': No such file or directory
ls: cannot access 'films/': No such file or directory
$ cat picture-and-movies.txt
img1.png
img2.png
img3.png
terror-on-elm-st1.avi
terror-on-elm-st2.avi
terror-on-elm-st3.avi
```

The streams have numbers

- 0 - standard in
- 1 - standard out
- 2 - standard err

You can redirect standard out to standard err

- When you write scripts and programs, it's a good habit to write error messages to std err (that's what people expect!)
- In bash, you print with echo, which uses std out by default
- To echo to std err instead, you can do this:
echo "Something went wrong." >&2
- The number 2 stands for std err
- We apologize for this syntax, it isn't pretty but it works