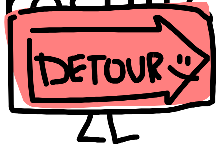
BIF703

stdin, stdout, stderr
Redirection

stdin, stdout, stderr

- Recall the Unix philosophy “do one thing well”.
- Unix has over one thousand commands (utilities) to perform a specific task.
- Although these specific commands may not be powerful alone, a Unix tool called “redirection” can be used to achieve very powerful results.



stdin, stdout, stderr

In order to understand the Unix tool of redirection, you must first understand what can be redirected, and then how it can be redirected.

- STDIN - Standard Input
  - Data read from a file or terminal (eg. keyboard)
- STDOUT - Standard Output
  - Data (output) as a result of a command or program executed.
- STDERR - Standard Error
  - Error message as a result of improper syntax of command or factors that lead to failure of task.

stdin

STDIN - Standard Input

- Data read from a file or terminal (eg. keyboard)
- You can already used stdin, but probably take it for granted when you learned to issue common Linux commands such as cat, grep, and sort
- To truly appreciate how these commands use stdin, let's look at some examples on the next few slides...

stdin

- Take the following commands as an example:

```
cat a1.c
grep –i “ULI101” index.html
```

- The cat command automatically sends (redirects) the contents of the file a1.c into the **`cat`** command which displays the contents on the screen.

- The grep command automatically sends (redirects) the contents of the file index.html into the command **`grep –i`** “ULI101” which will display lines in that file that match the pattern “ULI101”

stdin

It seems strange, but you are re-learning the "mechanics" of the commands in terms of stdin!

Procedure (questions to ask yourself when issuing a command that accepts stdin):

- If a command accepts stdin (for example a command with a filename as an argument, or filename expansion), assuming the file exists, that content is redirected as stdin into the command.
- Next, ask yourself what the command do? In other words, "here is the content from stdin into the command – how does the command "do to it"?

stdin

- Here are the contents of the file numbers.txt:

```
cat numbers.txt
101
1
20
```

- Look at the steps:

```
sort –n numbers.txt
```

stdin

- Here are the contents of the file numbers.txt:

```
cat numbers.txt
101
1
20
```

- Look at the steps:

```
sort –n numbers.txt
```

STEP 1:

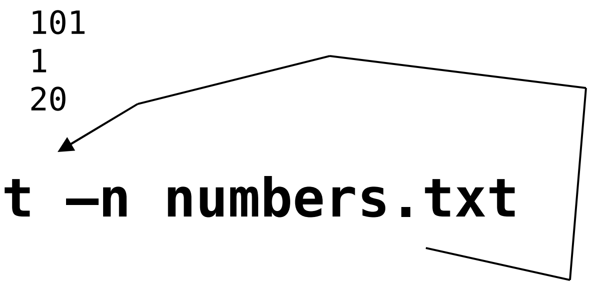
Visualize the contents of file numbers.txt being redirected as stdin into the command `sort –n`

stdin

- Here are the contents of the file numbers.txt:

```
cat numbers.txt
101
1
20
```

- Look at the steps:



```
101
1
20
```

```
sort –n numbers.txt
```

STEP 1:

Visualize the contents of file numbers.txt being redirected as stdin into the command `sort –n`

stdin

- Here are the contents of the file numbers.txt:

```
cat numbers.txt
101
1
20
```

- Look at the steps:

```
101
1
20
```

`sort –n numbers.txt`

STEP 2:

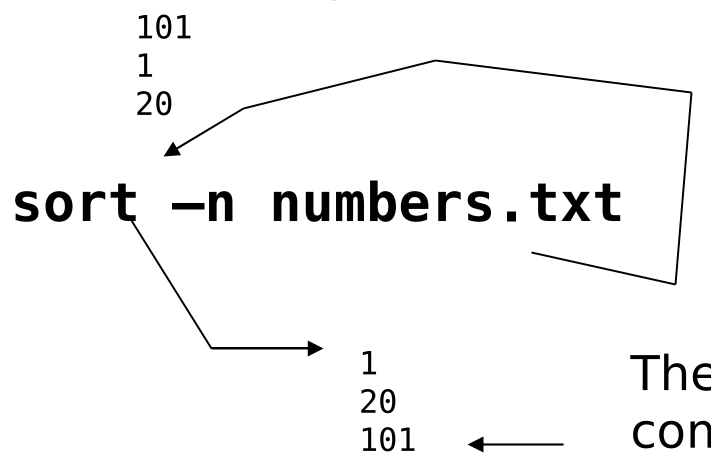
Ask the question, “what will this command (with option) do to the stdin?

stdin

- Here are the contents of the file numbers.txt:

```
cat numbers.txt
101
1
20
```

- Look at the steps:



```
101
1
20
```

```
sort –n numbers.txt
```

```
1
20
101
```

STEP 2:

Ask the question, “what will this command (with option) do to the stdin?

The result is from what the command did to the *stdin* is called the stdout. We discuss this in the next slide.

stdin

Question:

- If you issue the cat command without a filename (i.e. no arguments), what happens?
- Can you explain what is happening here in terms of STDIN and STDOUT?
- By the way, to get out of this “nightmare”, just kill the current process by pressing CTRL-c keys.

stdout

STDOUT - Standard Output

- Data (output) as a result of a command or program executed.

- You may have learned from the previous slide that if many commands like cat, sort, and grep require **stdin**, and if no filename is provided it gets it from the keyboard.

- The same applies to STDOUT: by default, the output from a command will be sent (or redirected) to the terminal's screen

stderr

STDERR - Standard Error

- Error message as a result of improper syntax of command or factors that lead to failure of task.

- What happens if a command like cat, sort, or grep uses a non-existent filename for its argument? An error message would be displayed. This error message is referred to as STDERR, and it is sent to the terminal's screen by default

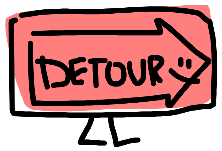
Exercise

Here is a good exercise to see if you really understand stdin, stdout, stderr:

- Assume there are only two files in your current directory: A and C. The file A only contains the text “this is file A”, and the file C only contains the text “this is file C”.
- Explains what happens in terms of stdin, stdout and stderr in the following command is issued:

```
cat A  B  C
```

Redirection



- OK, so I learned how to re-learn how command such as cat, sort, and grep in terms of STDIN, STDOUT, and STDERR… Big Deal …
- It’s a very BIG DEAL – learning this provides a foundation to do some pretty powerful things.
- You see, in UNIX/LINUX everything is a file, so in stead of redirecting in terms of a terminal, you can redirect to and from files. This skill allows people to actually write programs (script files) to do very complex operations involving stdin, stdout, stderr redirection!

Redirecting STDIN

<

Redirects standard input from a file to a Unix command.

Example:

```
mail username@learn < myfile
```

(i.e. a cool way to send a file as an attachment...)

Redirecting STDOUT

1> or > Redirects standard output to a file. (Will delete or “overwrite” any existing contents in the file).

Example:

- ls > listing.txt

1>> or >> Redirects standard output to a file but adds to the bottom of file’s existing contents.

Example:

- cat work >> things_to_do

Redirecting STDERR

2> Redirects the standard error to a file. This can be used to write error messages to a file for later reference.
2>> appends error message to bottom of existing file.

Example:

```
 cat  a  b  c 2> error_file
```

(Note: if file “b” does not exist, error message is redirected to file called “error file”.)

Exercise

- Assume that the only file in the directory is A and C with same contents in each file as in the previous Exercise.

- What will appear on the terminal screen if the following command is issued?

```
sort >> me 2> you < A  B  C
```

- By the way, don’t panic, work it out in terms of STDIN, STDOUT and STDERR!

Additional Resources

- Here are some Related-Links for Interest Only:

  Redirection of stdin, stdout and stderr
  - http://www.december.com/unix/tutor/redirect.html