The Unix Shell

Pipes and Filters

Copyright © Software Carpentry 2010
This work is licensed under the Creative Commons Attribution License
See http://software-carpentry.org/license.html for more information.
shell
shell

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>pwd</strong></td>
<td><strong>mkdir</strong></td>
</tr>
<tr>
<td><strong>cd</strong></td>
<td><strong>nano</strong></td>
</tr>
<tr>
<td><strong>ls</strong></td>
<td><strong>rm</strong></td>
</tr>
<tr>
<td><strong>.</strong></td>
<td><strong>rmdir</strong></td>
</tr>
<tr>
<td><strong>..</strong></td>
<td><strong>mv</strong></td>
</tr>
<tr>
<td></td>
<td><strong>cp</strong></td>
</tr>
</tbody>
</table>
shell

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pwd</td>
<td>directory name</td>
</tr>
<tr>
<td>mkdir</td>
<td>make directory</td>
</tr>
<tr>
<td>cd</td>
<td>change directory</td>
</tr>
<tr>
<td>nano</td>
<td>text editor</td>
</tr>
<tr>
<td>ls</td>
<td>list files</td>
</tr>
<tr>
<td>rm</td>
<td>remove file</td>
</tr>
<tr>
<td>rmdir</td>
<td>remove directory</td>
</tr>
<tr>
<td>mv</td>
<td>move file</td>
</tr>
<tr>
<td>cp</td>
<td>copy file</td>
</tr>
</tbody>
</table>

More powerful when combined
$ ls molecules

cubane.pdb    ethane.pdb    methane.pdb
octane.pdb    pentane.pdb    propane.pdb
$

Introduction
$ ls molecules

cubane.pdb    ethane.pdb    methane.pdb
octane.pdb    pentane.pdb    propane.pdb

$ cd molecules

$
$ ls molecules

cubane.pdb    ethane.pdb    methane.pdb
octane.pdb    pentane.pdb    propane.pdb

$ cd molecules

$ wc *.pdb  * is a wild card
$ ls molecules
  cubane.pdb  ethane.pdb  methane.pdb
  octane.pdb  pentane.pdb  propane.pdb
$ cd molecules
$ wc *.pdb

* is a wild card
matches zero or more characters
$ ls molecules
cubane.pdb  ethane.pdb  methane.pdb
octane.pdb  pentane.pdb  propane.pdb
$ cd molecules
$ wc *.pdb

* is a wild card matches zero or more characters so *.pdb matches all filenames ending in .pdb
$ ls molecules
cubane.pdb    ethane.pdb    methane.pdb
octane.pdb    pentane.pdb   propane.pdb
$ cd molecules
$ wc *.pdb
wc cubane.pdb ethane.pdb methane.pdb octane.pdb pentane.pdb propane.pdb
$ ls molecules
  cubane.pdb   ethane.pdb   methane.pdb
  octane.pdb   pentane.pdb   propane.pdb

$ cd molecules

$ wc *.pdb  
  word count
$ ls molecules
  cubane.pdb  ethane.pdb  methane.pdb
  octane.pdb  pentane.pdb  propane.pdb
$ cd molecules
$ wc *.pdb

word count
counts lines, words, and characters in files
$ ls molecules
cubane.pdb ethane.pdb methane.pdb
octane.pdb pentane.pdb propane.pdb
$ cd molecules
$ wc *.pdb
 20  156 1158 cubane.pdb
 12   84  622 ethane.pdb
   9   57  422 methane.pdb
 30  246 1828 octane.pdb
 21  165 1226 pentane.pdb
 15  111  825 propane.pdb
107  819 6081 total
$
```bash
$ wc -l *.pdb  # report only lines lines
  20  cubane.pdb
  12  ethane.pdb
   9  methane.pdb
  30  octane.pdb
  21  pentane.pdb
  15  propane.pdb
 107  total
$
```
$ wc -l *.pdb  
 20  cubane.pdb  
 12  ethane.pdb  
  9   methane.pdb  
 30  octane.pdb  
 21  pentane.pdb  
 15  propane.pdb  
 107  total  
$

report only lines
use -w for words or
-c for characters
<table>
<thead>
<tr>
<th>File</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>cubane.pdb</td>
<td>20</td>
</tr>
<tr>
<td>ethane.pdb</td>
<td>12</td>
</tr>
<tr>
<td>methane.pdb</td>
<td>9</td>
</tr>
<tr>
<td>octane.pdb</td>
<td>30</td>
</tr>
<tr>
<td>pentane.pdb</td>
<td>21</td>
</tr>
<tr>
<td>propane.pdb</td>
<td>15</td>
</tr>
</tbody>
</table>

Which file is shortest?

Pipes and Filters

Introduction
Which file is shortest?

Easy to see when there are six...

20  cubane.pdb
12  ethane.pdb
 9  methane.pdb
30  octane.pdb
21  pentane.pdb
15  propane.pdb
107  total
20  cubane.pdb
12  ethane.pdb
 9  methane.pdb
30  octane.pdb
21  pentane.pdb
15  propane.pdb
107  total

Which file is shortest?
Easy to see when there are six...
...but what if there were 6000?
$ wc -l *.pdb > lengths
$

$ wc -l *.pdb > lengths
$ 

*redirect* output to a file
$ wc -l *.pdb > lengths
$ 

*redirect* output to a file
create file if it doesn't exist
$ wc -l *.pdb > lengths
$

redirect output to a file
create file if it doesn't exist
overwrite it if it does
$ wc -l *.pdb > lengths
$

no screen output
$ wc -l *.pdb > lengths
$ ls lengths

lengths

$
```bash
$ wc -l *.pdb > lengths
$ ls lengths
lengths
$ cat lengths
   20  cubane.pdb
   12  ethane.pdb
    9  methane.pdb
   30  octane.pdb
   21  pentane.pdb
   15  propane.pdb
   107 total
$
```

Pipes and Filters

Introduction
$ wc -l *.pdb > lengths
$ ls lengths
lengths
$ cat lengths
20  cubane.pdb
12  ethane.pdb
 9  methane.pdb
30  octane.pdb
21  pentane.pdb
15  propane.pdb
107  total
$

concatenate files

Pipes and Filters
Introduction
```bash
$ wc -l *.pdb > lengths
$ ls lengths
lengths
$ cat lengths
  20  cubane.pdb
  12  ethane.pdb
   9  methane.pdb
  30  octane.pdb
  21  pentane.pdb
  15  propane.pdb
 107  total
$ 
```

**concatenate files**
in this case, only one
so file contents printed to screen
$ sort lengths
  9  methane.pdb
  12  ethane.pdb
  15  propane.pdb
  20  cubane.pdb
  21  pentane.pdb
  30  octane.pdb
  107  total
$

Pipes and Filters

Introduction
$ sort lengths > sorted-lengths

$
$ sort lengths > sorted-lengths
$ head -1 sorted-lengths
  9  methane.pdb
$
sort lengths > sorted-lengths

head -1 sorted-lengths

9 methane.pdb

get the first line of the file
$ sort lengths > sorted-lengths
$ head -1 sorted-lengths
9 methane.pdb

get the first line of the file
this must be the PDB file
with the fewest lines,
since sorted-lengths holds
files and line counts in
order from least to greatest
\$ \text{sort lengths} > \text{sorted-lengths}
\$
$\text{head -1 sorted-lengths}$

9 \textit{methane.pdb}

not particularly obvious

get the first line of the file this must be the PDB file with the fewest lines, since sorted-lengths holds files and line counts in order from least to greatest

\textbf{Pipes and Filters}

\textbf{Introduction}
```bash
$ sort lengths | head -1
 9  methane.pdb
$
```
$ sort lengths | head -1
  9  methane.pdb

$ a pipe
$ sort lengths | head -1
9  methane.pdb

$a pipe$
use output of left side
$ sort lengths | head -1

9  methane.pdb

$ a pipe
use output of left side
as input to right side
$ sort lengths | head -1
  9  methane.pdb
$

a pipe
use output of left side
as input to right side
without creating temporary file
$ wc -l *.pdb | sort | head -1

9  methane.pdb

$ don't need to create lengths file
$ wc -l *.pdb | sort | head -1
   9  methane.pdb
$

This simple idea is why Unix has been so successful
This simple idea is why Unix has been so successful
Create simple tools that:
This simple idea is why Unix has been so successful
Create simple tools that:
– do one job well
This simple idea is why Unix has been so successful: Create simple tools that:
- do one job well
- work well with each other
```bash
$ wc -l *.pdb | sort | head -1
  9  methane.pdb
$ 
```

This simple idea is why Unix has been so successful. Create simple tools that:
- do one job well
- work well with each other

10 tools can be combined in 100 ways.
running program
running program

process
standard input
standard output stdout
$ wc -l *.pdb > lengths
$ wc -l *.pdb > lengths
Introduc$$

```bash
$ wc -l *.pdb > lengths
```
Introduction

$ wc -l *.pdb > lengths
Pipes and Filters

Introduction

```bash
$ wc -l *.pdb | sort
```
$ wc -l *.pdb | sort | head -1
This programming model called *pipes and filters*
This programming model called *pipes and filters*

A *filter* transforms a stream of input into a stream of output
This programming model called *pipes and filters*

A *filter* transforms a stream of input into a stream of output

A *pipe* connects two filters
This programming model called *pipes and filters*

A *filter* transforms a stream of input into a stream of output.

A *pipe* connects two filters.

Any program that reads lines of text from standard input, and writes lines of text to standard output, can work with every other.
This programming model called *pipes and filters*

A *filter* transforms a stream of input into a stream of output

A *pipe* connects two filters

Any program that reads lines of text from standard input, and writes lines of text to standard output, can work with every other

You can (and should) write such programs
<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pwd</code></td>
<td>mkdir</td>
</tr>
<tr>
<td><code>cd</code></td>
<td>nano</td>
</tr>
<tr>
<td><code>ls</code></td>
<td>rm</td>
</tr>
<tr>
<td>.</td>
<td>rmdir</td>
</tr>
<tr>
<td>..</td>
<td>mv</td>
</tr>
<tr>
<td></td>
<td>cp</td>
</tr>
<tr>
<td>Command</td>
<td>Command</td>
</tr>
<tr>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td><code>pwd</code></td>
<td><code>mkdir</code></td>
</tr>
<tr>
<td><code>cd</code></td>
<td><code>nano</code></td>
</tr>
<tr>
<td><code>ls</code></td>
<td><code>rm</code></td>
</tr>
<tr>
<td><code>.</code></td>
<td><code>rmdir</code></td>
</tr>
<tr>
<td><code>..</code></td>
<td><code>mv</code></td>
</tr>
<tr>
<td></td>
<td><code>cp</code></td>
</tr>
</tbody>
</table>

**Pipes and Filters**

**Introduction**
<table>
<thead>
<tr>
<th>Command</th>
<th>Command</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>pwd</td>
<td>mkdir</td>
<td>wc</td>
</tr>
<tr>
<td>cd</td>
<td>nano</td>
<td>sort</td>
</tr>
<tr>
<td>ls</td>
<td>rm</td>
<td>head</td>
</tr>
<tr>
<td>.</td>
<td>rmdir</td>
<td>tail</td>
</tr>
<tr>
<td>..</td>
<td>mv</td>
<td>split</td>
</tr>
<tr>
<td>cp</td>
<td>cut</td>
<td>uniq</td>
</tr>
<tr>
<td>Command</td>
<td>Command</td>
<td>Command</td>
</tr>
<tr>
<td>---------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td><code>pwd</code></td>
<td><code>mkdir</code></td>
<td><code>wc</code></td>
</tr>
<tr>
<td><code>cd</code></td>
<td><code>nano</code></td>
<td><code>sort</code></td>
</tr>
<tr>
<td><code>ls</code></td>
<td><code>rm</code></td>
<td><code>head</code></td>
</tr>
<tr>
<td>.</td>
<td><code>rmdir</code></td>
<td><code>tail</code></td>
</tr>
<tr>
<td>..</td>
<td><code>mv</code></td>
<td><code>split</code></td>
</tr>
<tr>
<td></td>
<td><code>cp</code></td>
<td><code>cut</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>uniq</code></td>
</tr>
</tbody>
</table>

### Pipes and Filters

**Introduction**


<table>
<thead>
<tr>
<th>Command</th>
<th>Command</th>
<th>Command</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pwd</code></td>
<td><code>mkdir</code></td>
<td><code>wc</code></td>
<td><code>*</code></td>
</tr>
<tr>
<td><code>cd</code></td>
<td><code>nano</code></td>
<td><code>sort</code></td>
<td><code>&gt;</code></td>
</tr>
<tr>
<td><code>ls</code></td>
<td><code>rm</code></td>
<td><code>head</code></td>
<td>`</td>
</tr>
<tr>
<td><code>.</code></td>
<td><code>rmdir</code></td>
<td><code>tail</code></td>
<td><code>&lt;</code></td>
</tr>
<tr>
<td><code>..</code></td>
<td><code>mv</code></td>
<td><code>split</code></td>
<td><code>?</code></td>
</tr>
<tr>
<td><code>cp</code></td>
<td><code>cut</code></td>
<td><code>uniq</code></td>
<td></td>
</tr>
</tbody>
</table>

**Pipes and Filters**

**Introduction**