

Reusable software - the UNIX way

KDE Conf India, 2015

Noufal Ibrahim

17 April 2015



Thank you!

- Thank you for putting together such a nice event.



Thank you!

- Thank you for putting together such a nice event.
- Thank you for doing it in Kerala.



Thank you!

- Thank you for putting together such a nice event.
- Thank you for doing it in Kerala.
- and thank you for inviting me to present here.



Confessions



Confessions

- I am not a KDE user.

Confessions

- I am not a KDE user.
- I was a GNOME user.



Confessions

- I am not a KDE user.
- I was a GNOME user.
- I use a few graphical applications.

Confessions

- I am not a KDE user.
- I was a GNOME user.
- I use a few graphical applications.
- But I'm a big fan of the command line.



Confessions

- And this is a presentation on why you should be too.



Confessions

- And this is a presentation on why you should be too.
- So get ready.



UNIX philosophy

This is the Unix philosophy: Write programs that do one thing and do it well. Write programs to work together. Write programs to handle text streams, because that is a universal interface.

– Doug McIlroy



Building blocks

- Gives you tools rather than products.

Building blocks

- Gives you tools rather than products.
- Just add imagination!

Building blocks

- Gives you tools rather than products.
- Just add imagination!
- Small, well debugged, powerful little utilities.



Building blocks

- Gives you tools rather than products.
- Just add imagination!
- Small, well debugged, powerful little utilities.
- All communicate via. a common interface - text streams.

Basic ideas

- Programs produce and consume textual output

Basic ideas

- Programs produce and consume textual output
- Let's look at a few examples



`ls` lists files in the current directory.

ls lists files in the current directory.

```
ls
```



`ls` lists files in the current directory.

```
ls
```

→

```
poem.txt  fruits.txt
```

cat outputs a file

```
cat poem.txt
```



cat outputs a file

```
cat poem.txt
```

→

```
The secret of wisdom?  
It's plain you see,  
It's easy to express
```

```
to err and err and err again  
but less and less and less
```



wc counts characters, words and lines in files

```
wc poem.txt
```



wc counts characters, words and lines in files

```
wc poem.txt
```

→

```
6 25 120 poem.txt
```

`grep` searches for strings in files

```
grep express poem.txt
```



grep searches for strings in files

```
grep express poem.txt
```

→

It's easy to express



`sort` sorts lines of a file

```
sort poem.txt
```



sort sorts lines of a file

```
sort poem.txt
```

→

```
but less and less and less  
It's easy to express  
It's plain you see,  
The secret of wisdom?  
to err and err and err again
```

uniq shows unique lines of a sorted file

```
cat fruits.txt
```

```
Apples  
Apples  
Bananas  
Oranges  
Grapes  
Grapes  
Grapes
```



uniq shows unique lines of a sorted file

```
cat fruits.txt
```

```
Apples  
Apples  
Bananas  
Oranges  
Grapes  
Grapes  
Grapes
```

```
uniq -c fruits.txt
```

→

```
2 Apples  
1 Bananas  
1 Oranges  
3 Grapes
```

head **and** tail show top and bottom of files

head and tail show top and bottom of files

```
head -2 poem.txt
```

→

```
The secret of wisdom?  
It's plain you see,
```



head and tail show top and bottom of files

```
head -2 poem.txt
```

→

```
The secret of wisdom?  
It's plain you see,
```

```
tail -2 poem.txt
```

→

```
to err and err and err again  
but less and less and less
```



`awk` allows you to manipulate columns of a file

```
awk '{print $1}' poem.txt
```



awk allows you to manipulate columns of a file

```
awk '{print $1}' poem.txt
```

→

The
It's
It's

to
but

paste allows you to to combine multiple files as
columns

```
paste poem.txt fruits.txt
```



paste allows you to to combine multiple files as columns

```
paste poem.txt fruits.txt
```

→

```
The secret of wisdom?      Apples
It's plain you see,       Apples
It's easy to express      Bananas
                          Oranges
to err and err and err again  Grapes
but less and less and less  Grapes
                          Grapes
```

Combining commands

- The | character can be used to pipe the output of one command into the input of another.

Combining commands

- The | character can be used to pipe the output of one command into the input of another.

- e.g. `awk '{print $2}' poem.txt | sort` →

easy

err

less

plain

secret

One more. `tr` translates character sets

```
cat poem.txt | tr ' e' ' _.'
```

One more. `tr` translates character sets

```
cat poem.txt | tr ' e' '_.'
```

→

```
Th._s.cr.t_of_wisdom?
```

```
It'_s_plain_you_s..,
```

```
It'_s_.asy_to_.xpr.ss
```

```
to_.rr_and_.rr_and_.rr_again
```

```
but_l.ss_and_l.ss_and_l.ss
```



Now we have blocks

Let's build some buildings!



Now we have blocks

Let's build some buildings!

- I have a file called `moby-dick.txt`.



Now we have blocks

Let's build some buildings!

- I have a file called `moby-dick.txt`.
- I know that it's the text of a book.



Now we have blocks

Let's build some buildings!

- I have a file called `moby-dick.txt`.
- I know that it's the text of a book.
- I want to find out as much as I can about it using what we've seen so far.



Now we have blocks

Let's build some buildings!

- I have a file called `moby-dick.txt`.
- I know that it's the text of a book.
- I want to find out as much as I can about it using what we've seen so far.
- Available at

<http://www.gutenberg.org/cache/epub/2701/pg2701.txt>.



Let's do it live!



Let's do it live!

- Number of chapters.



Let's do it live!

- Number of chapters.
- Number of pages.



Let's do it live!

- Number of chapters.
- Number of pages.
- Sentence lengths.



Let's do it live!

- Number of chapters.
- Number of pages.
- Sentence lengths.
- Year of writing.

Let's do it live!

- Number of chapters.
- Number of pages.
- Sentence lengths.
- Year of writing.
- Statistically significant words.

Let's do it live!

- Number of chapters.
- Number of pages.
- Sentence lengths.
- Year of writing.
- Statistically significant words.
- Statistically significant proper nouns.

Let's do it live!

- Number of chapters.
- Number of pages.
- Sentence lengths.
- Year of writing.
- Statistically significant words.
- Statistically significant proper nouns.
- Statistically significant bigrams/trigrams.

Let's do it live!

- Number of chapters.
- Number of pages.
- Sentence lengths.
- Year of writing.
- Statistically significant words.
- Statistically significant proper nouns.
- Statistically significant bigrams/trigrams.
- Investigate the proper nouns.

Let's do it live!

- Number of chapters.
- Number of pages.
- Sentence lengths.
- Year of writing.
- Statistically significant words.
- Statistically significant proper nouns.
- Statistically significant bigrams/trigrams.
- Investigate the proper nouns.
- Make a summary.

What is *Moby Dick* about?

What is *Moby Dick* about?

Moby-Dick; or, *The Whale* (1851) is a novel by Herman Melville, in which the sole survivor of a lost **whaling ship** relates the tale of his **captain's** self-destructive obsession to hunt the **white whale**, Moby-Dick... The narrator, Ishmael,... his bunk mate, a heavily tattooed Polynesian **harpooner** named **Queequeg**... In **Nantucket**, the pair signs on with the **Pequod**, a **whaling ship**... The ship's **captain**, **Ahab**,... The chief mate is **Starbuck**, a serious, sincere Quaker and fine leader; second mate is **Stubb**,... Ahab secures their support for his single, secret purpose for this voyage: hunting down and killing Moby Dick, an old, very large sperm whale, with a snow-**white** hump and mottled skin, that crippled Ahab on his last whaling voyage.....

https://en.wikipedia.org/wiki/Moby_Dick



Take aways

- The command line has unique strengths.

Take aways

- The command line has unique strengths.
- If your building blocks are good, you can do almost magical things.

Take aways

- The command line has unique strengths.
- If your building blocks are good, you can do almost magical things.
- The command line is under appreciated and not sufficiently well known.

Take aways

- The command line has unique strengths.
- If your building blocks are good, you can do almost magical things.
- The command line is under appreciated and not sufficiently well known.
- Therefore, there are a lot of wheels being poorly reinvented.

Take aways

- The command line has unique strengths.
- If your building blocks are good, you can do almost magical things.
- The command line is under appreciated and not sufficiently well known.
- Therefore, there are a lot of wheels being poorly reinvented.
- Learn it well and take full advantage of it.

Thank you
and enjoy the conference!

noufal@nibrahim.net.in
<http://nibrahim.net.in>
@noufalibrahim

