

Unix

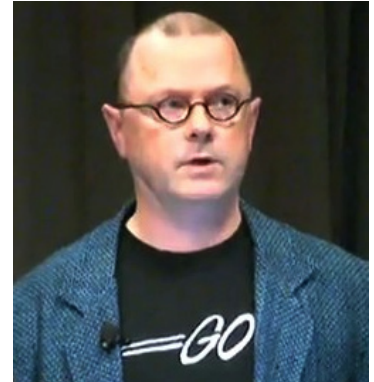
“Those who don’t understand Unix are condemned to reinvent it, poorly.”

Henry Spencer



“Unix never says ‘please’.”

Rob Pike





Ken Thompson and Dennis Ritchie

- Dates back to 1969.

- Dates back to 1969.





- Dates back to 1969.
- Written as a fun project, later developed into a patent application processing system.



- Dates back to 1969.
- Written as a fun project, later developed into a patent application processing system.
- Since then many variants were created, notably
 - GNU/Linux*
 - Android*
 - Darwin (macOS, iOS)*

After 50 years, UNIX is here to stay!

Hierarchical file system:

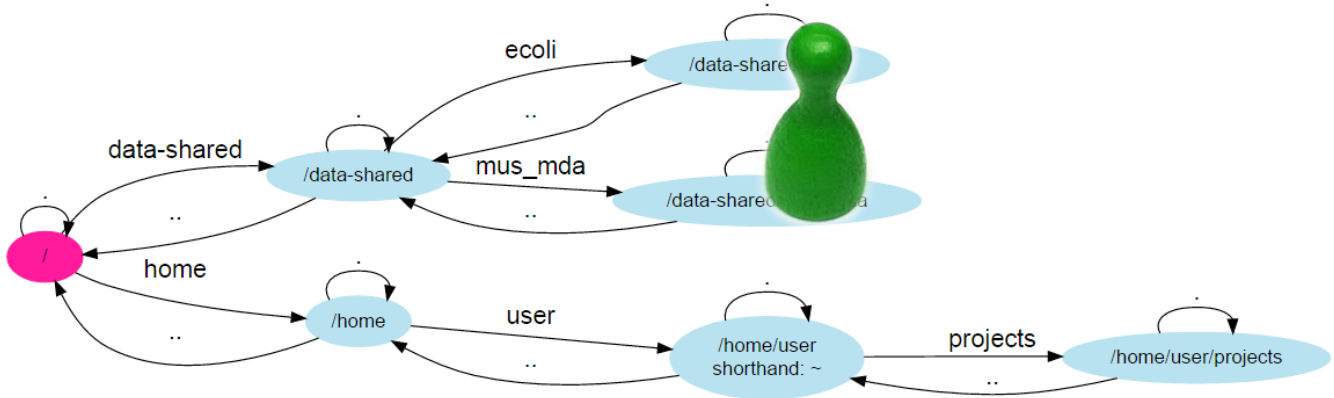
- everything is a file
- directories are used to organize files
- ***current working directory*** - “where I am”
shortcut: “.” (single dot)
- ***home directory*** - “where my stuff is”
shortcut: ‘~’ (tilde)
- ***root directory*** - “where it all begins”
name: ‘/’ (slash)

Paths are 'recipes' how to find a file or a directory.

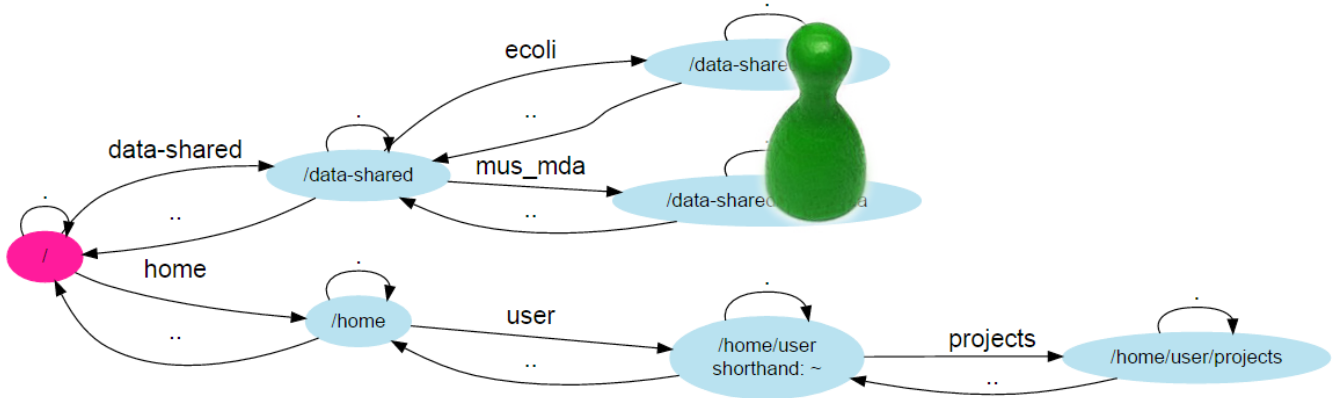
Path types:

- ***absolute path*** - *"how do I get there from Rome"*
starts with a '/' (root directory)
- ***relative path*** - *"how do I get there from here"*
path from current working directory
- ***'..'*** - relative path referring to 'one level up'

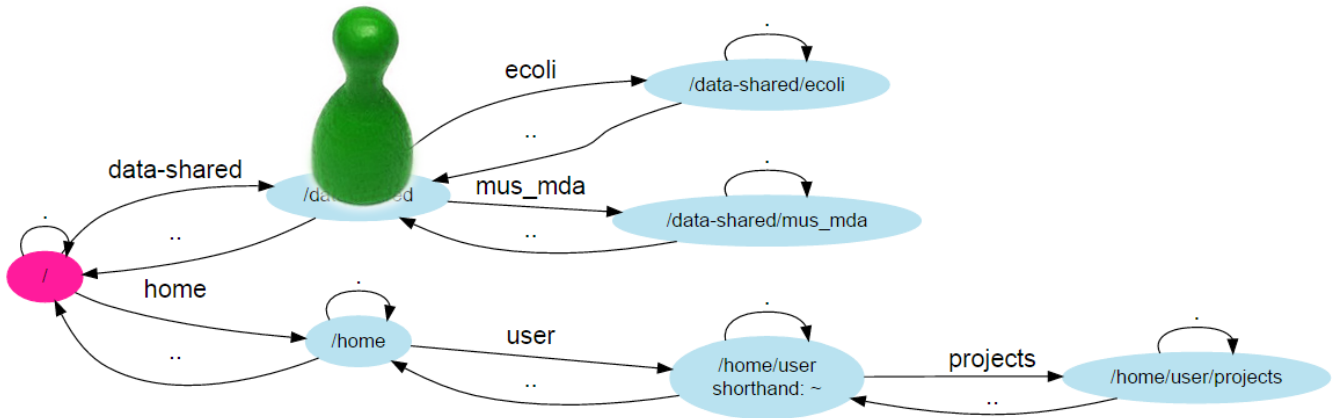
Why is '~/projects' an absolute path?



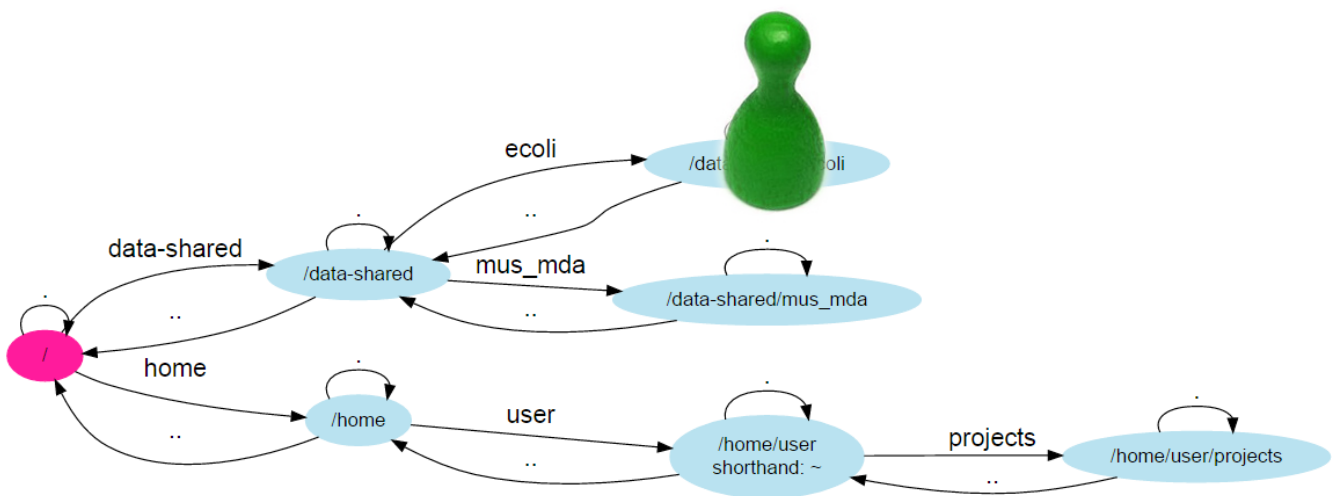
working directory



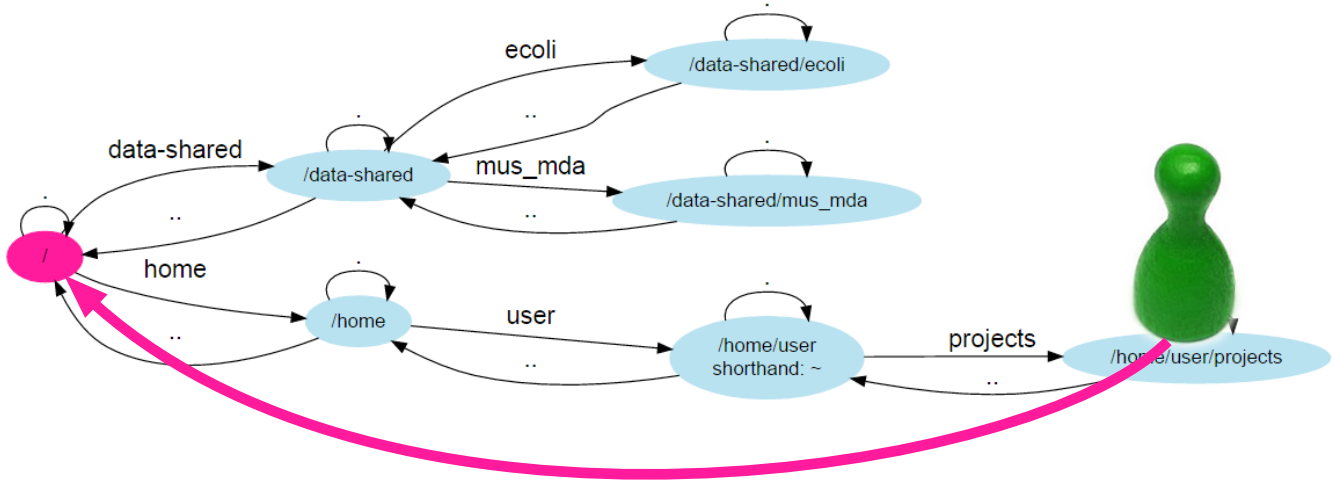
relative path: ../ecoli



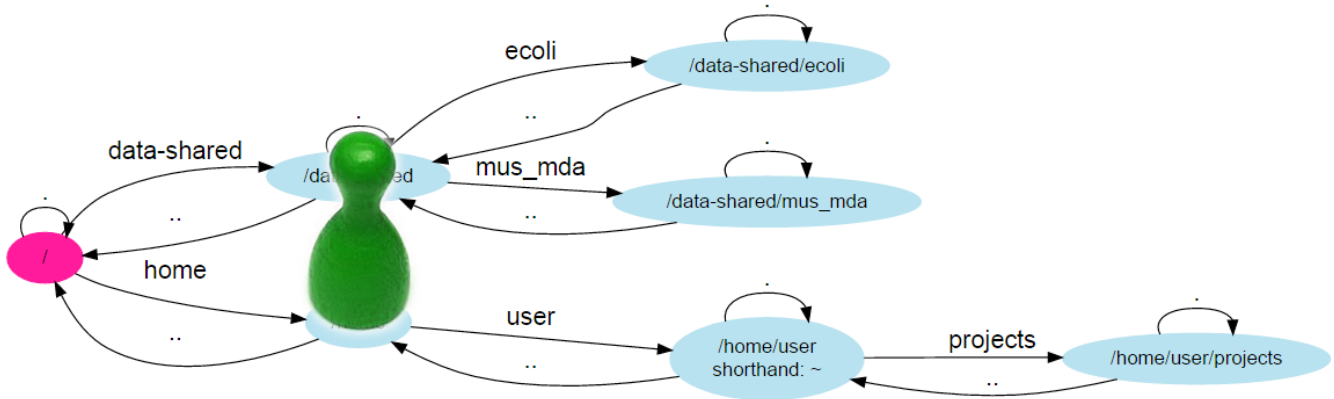
relative path: ../ecoli



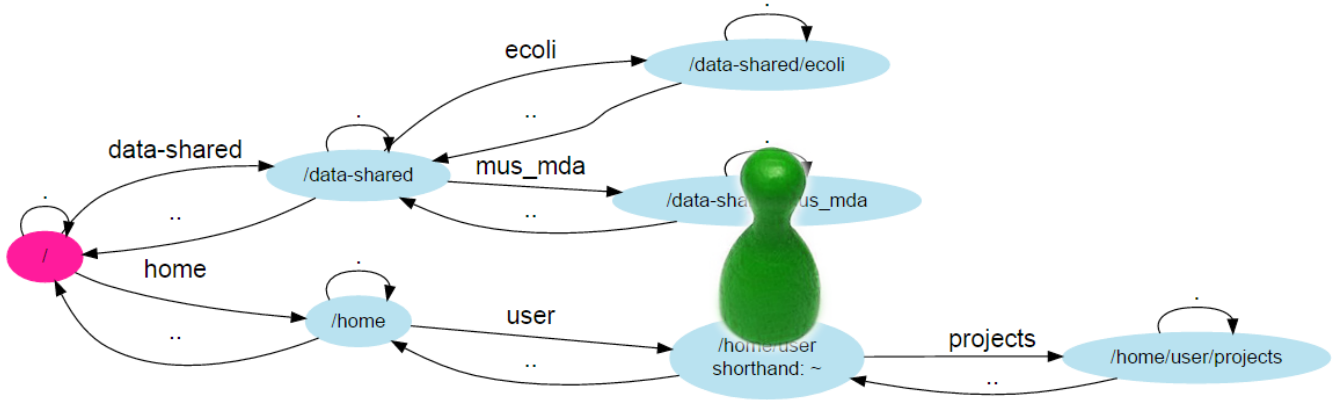
relative path: ../ecoli



absolute path: `/home/user`



absolute path: /home/user



absolute path: /home/user

Data is stored in files.

Tools read the data and produce new files.

Is there a way to combine more tools without storing all the intermediate results?

Pipes invented long before by Doug McIlroy.

Summary--what's most important.

To put my strongest concerns
into a nutshell:

1. We should have some ways of
coupling programs like garden
hose--screw in another segment when
it becomes when it becomes necessary
to massage data in another way.
This is the way of IO also.

...



M. D. McIlroy
October 11, 1964



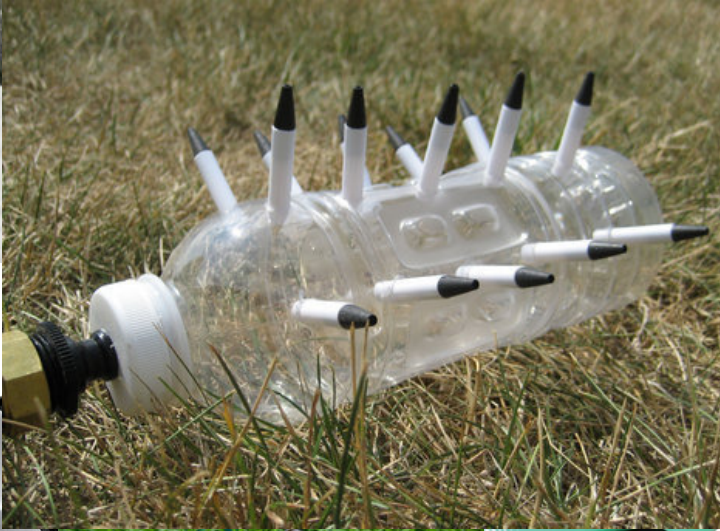












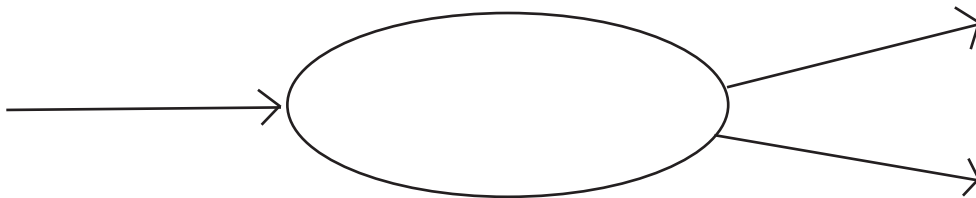


What is so good about UNIX?

- flexibility
- conciseness
- automation

Flexibility: every program has

- standard input
- standard output
- standard error

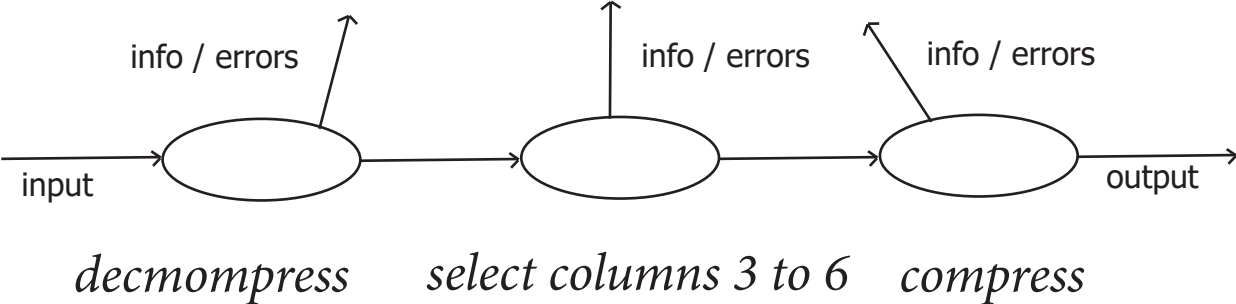


Flexibility: every “unix” program

- works with **text**
- reads data line by line
- outputs data line by line
- does one simple operation

Flexibility: programs can be chained

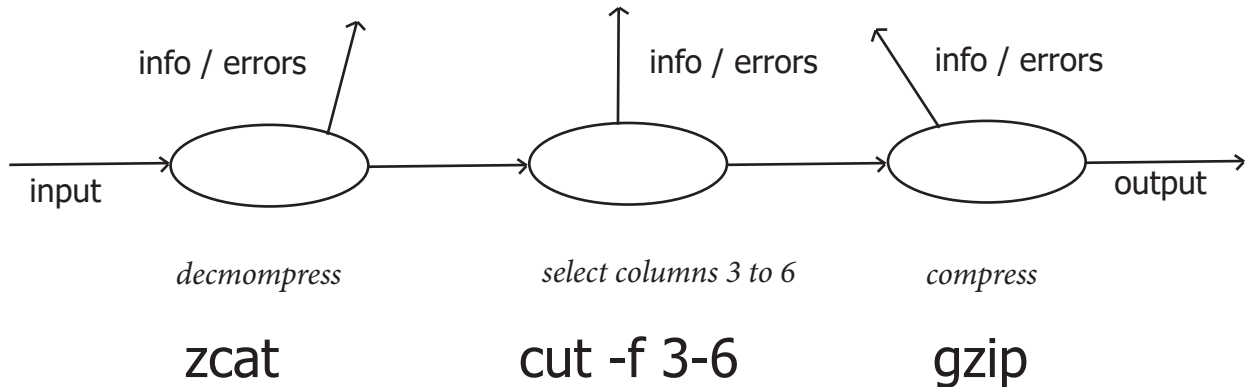
USER



Conciseness:

“with just 30 characters you can do almost anything”

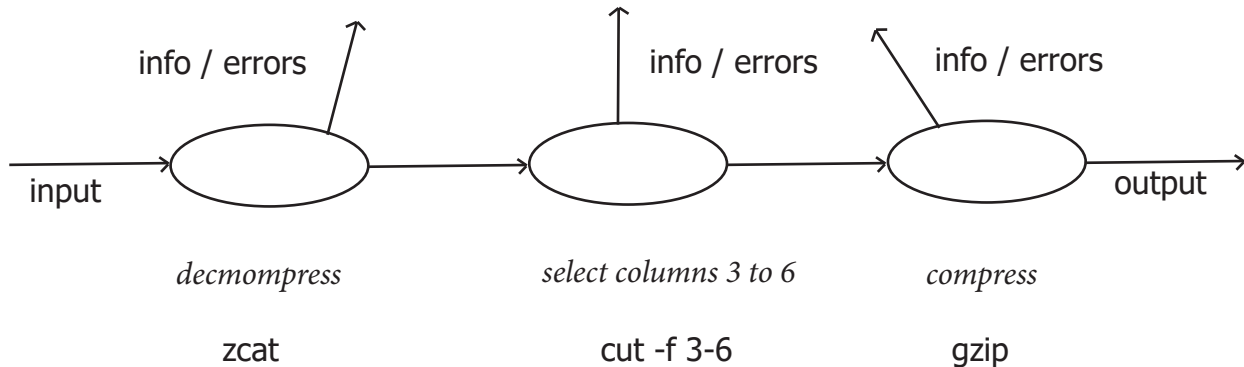
USER



Automation:

every operation can be stored as a text “recipe”

USER



<bigtable.gz **zcat** | **cut** -f 3-6 | **gzip** >bigtable-3-6.gz

Why is it so good for genomics?

And what is Linux then?

