Unix

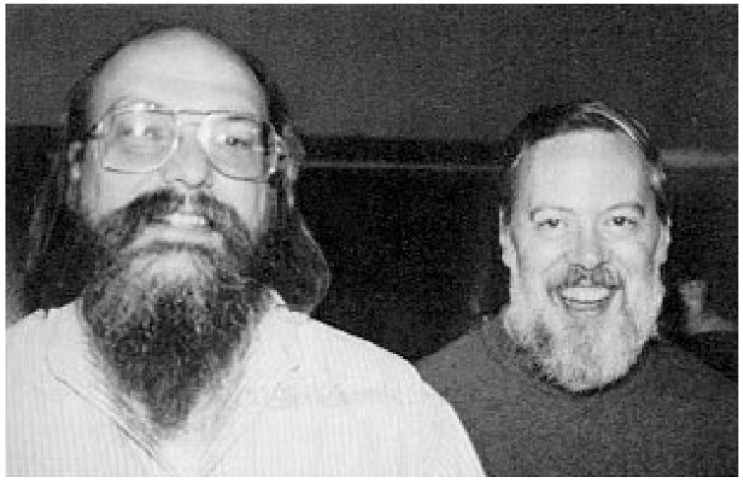
"Those who don't understand Unix are condemned to reinvent it, poorly."

Henry Spencer

“Unix never says ’please’.”

Rob Pike

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



Ken Thompson and Dennis Ritchie

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

After more than 50 years, UNIX is here to stay!

Design principles

- Hierarchical file system
- Data stored in files
- Tools read the data and produce new files

Hierarchical file system:

- everything is a file
- directories are used to organize files

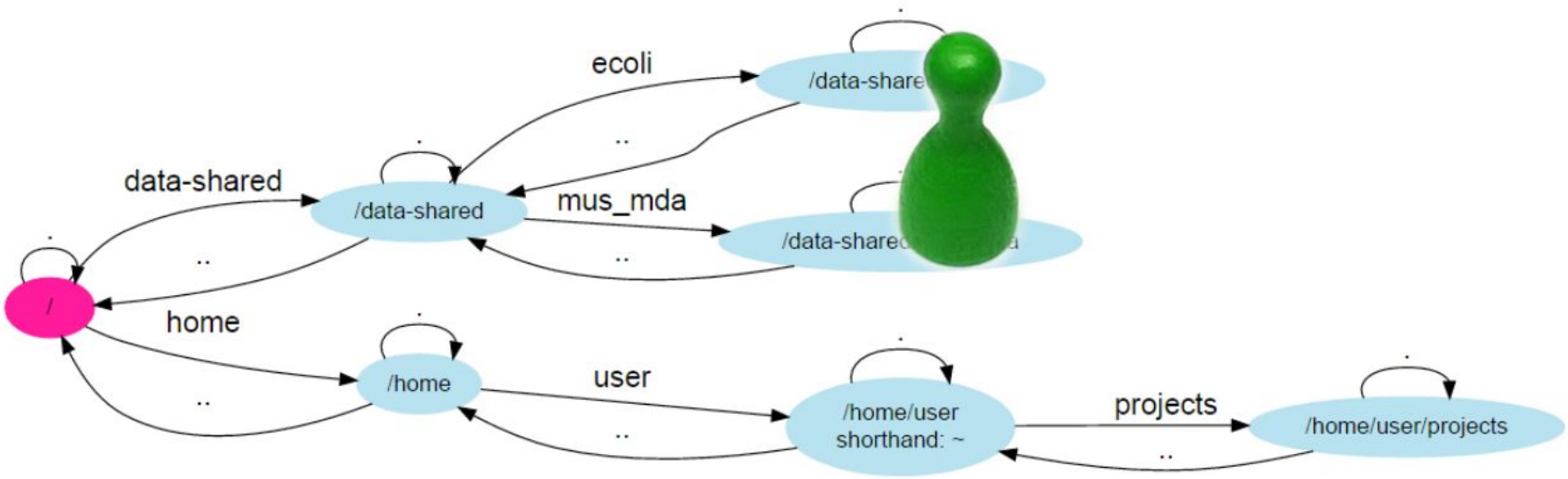
  - ***current working directory*** - *"where I am"*
    shortcut: '.' (single dot)
  - ***home directory*** - *"where my stuff is"*
    shortcut: '~' (tilde)
  - ***root directory*** - *"where it all begins"*
    name: '/' (slash)

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

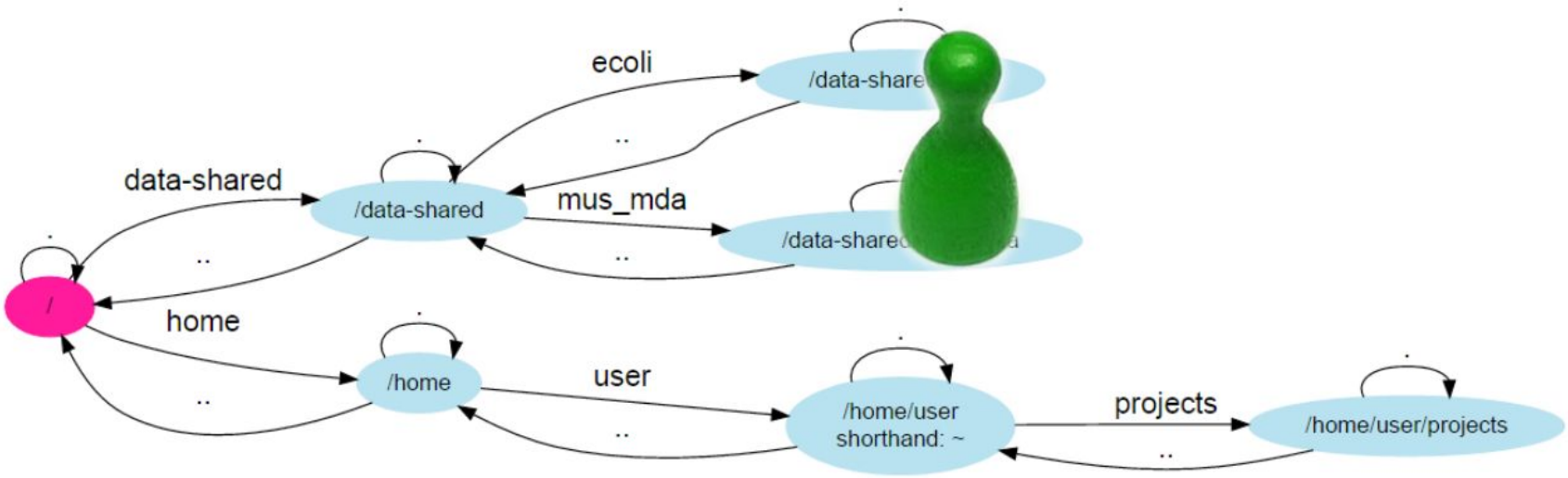
Path types:

- ***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'

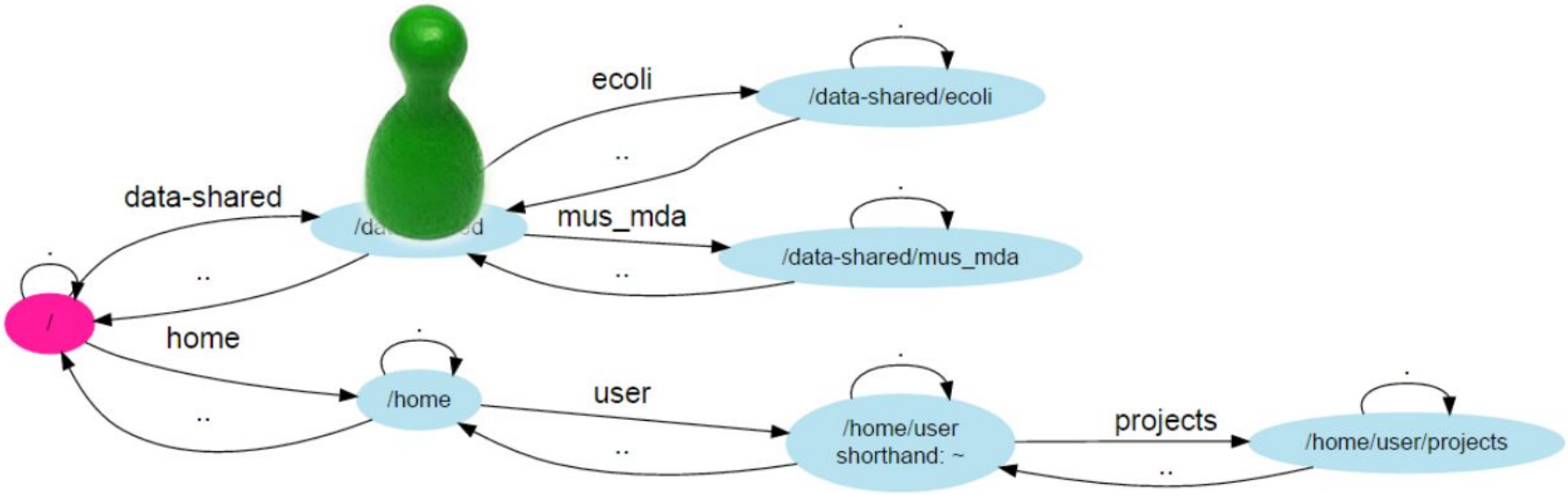
Why is '~/projects' an absolute path?



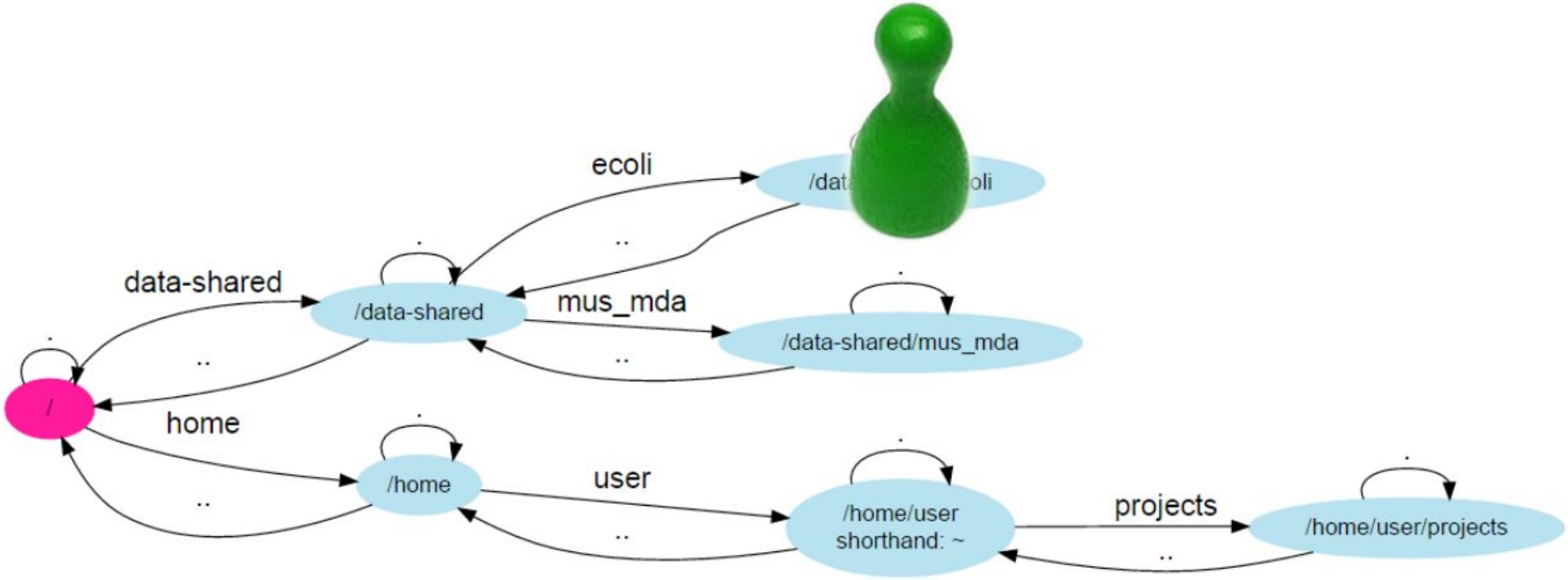
working directory



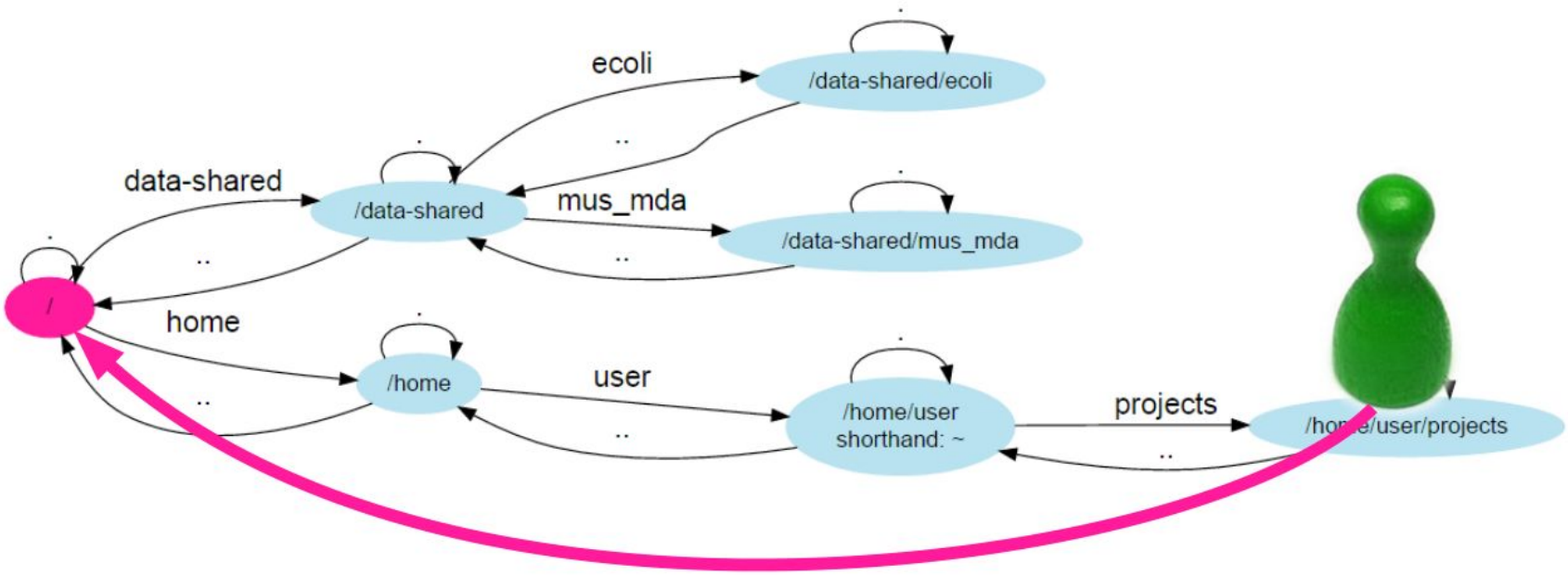
relative path: ../ecoli



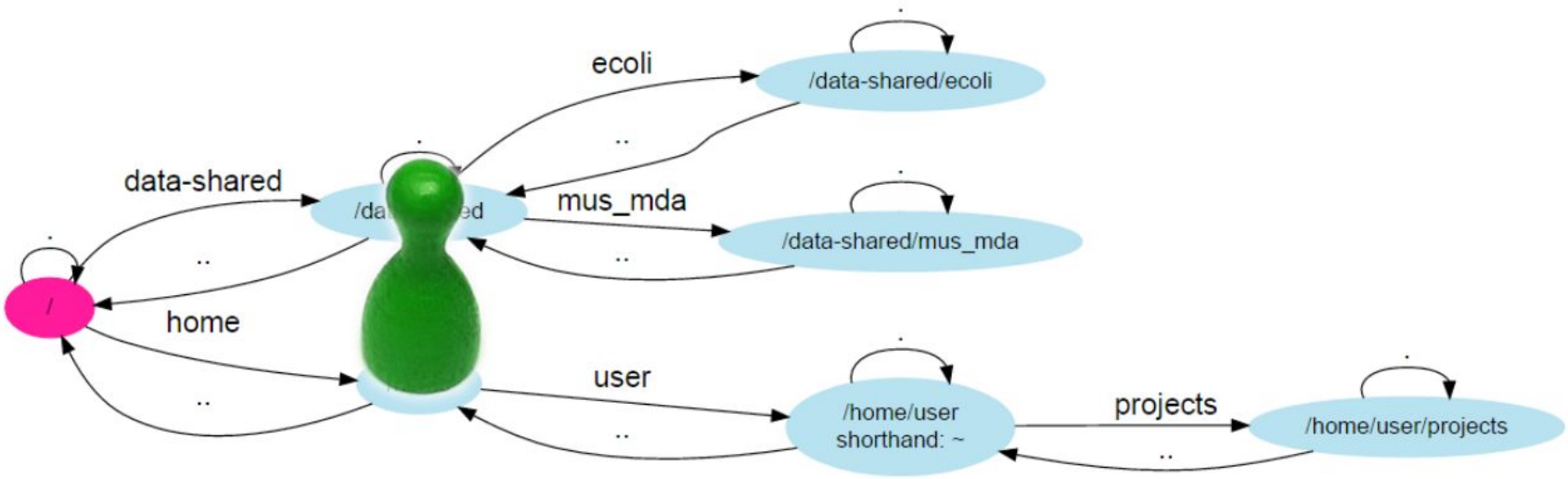
relative path: ../ecoli



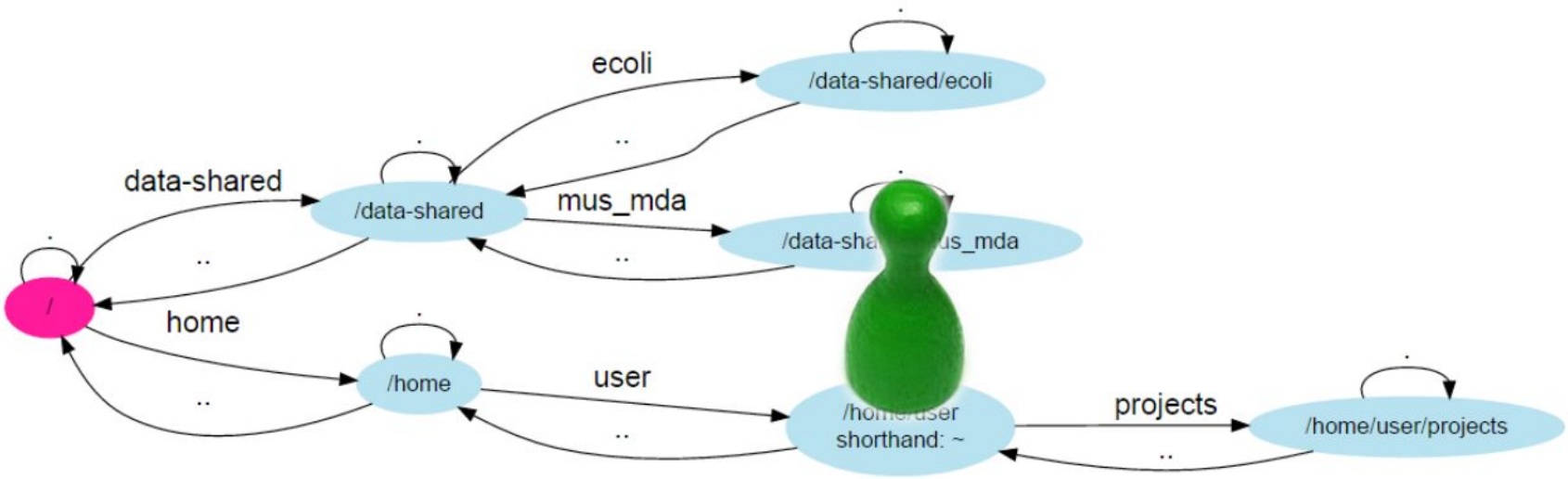
relative path: ../ecoli



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


FLEXOGEN


Gilmour FLEXOGEN


FLEXOGEN


FLEXOGEN


Gilmour FLEXOGEN

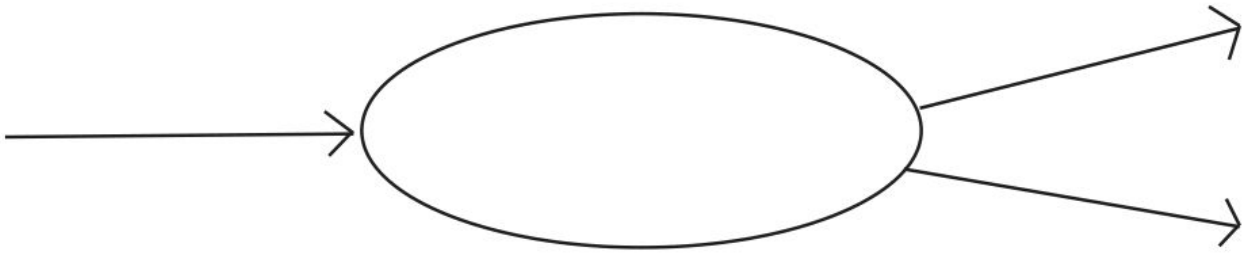

FLEXOGEN

What is so good about UNIX?

- flexibility
- conciseness
- automation

Flexibility: every program has

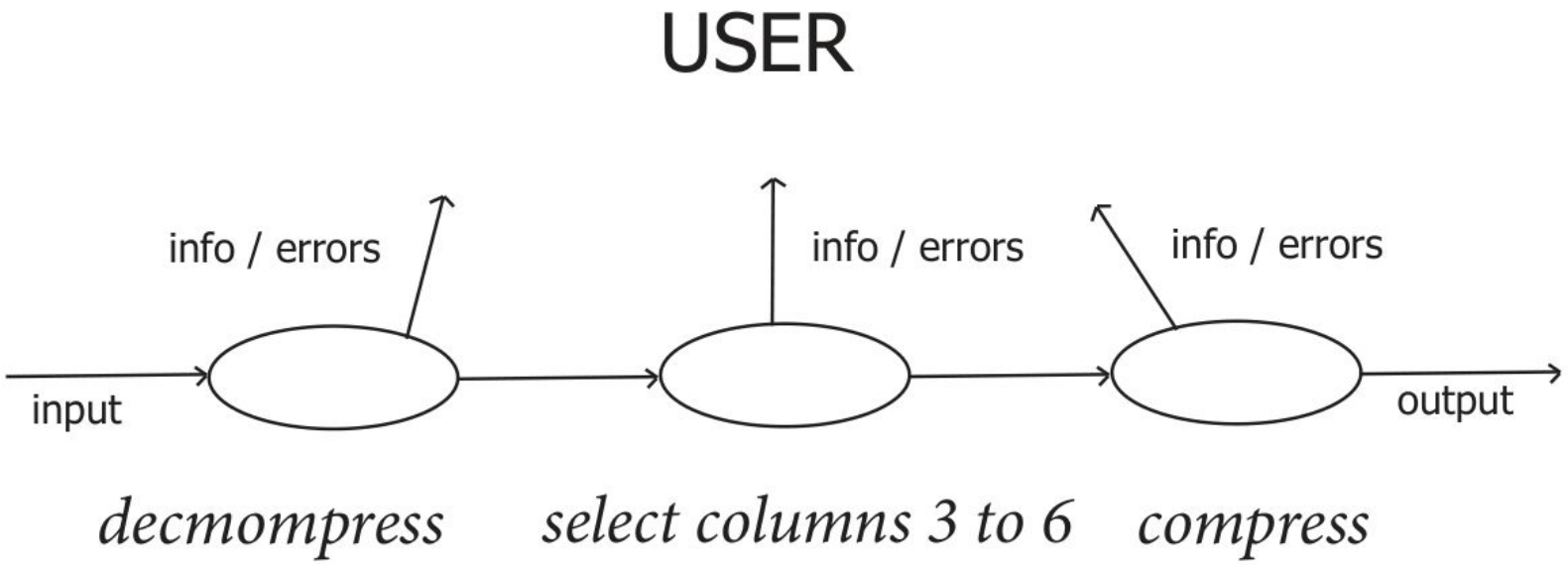
- standard input
- standard output
- standard error



Flexibility: every "unix" program

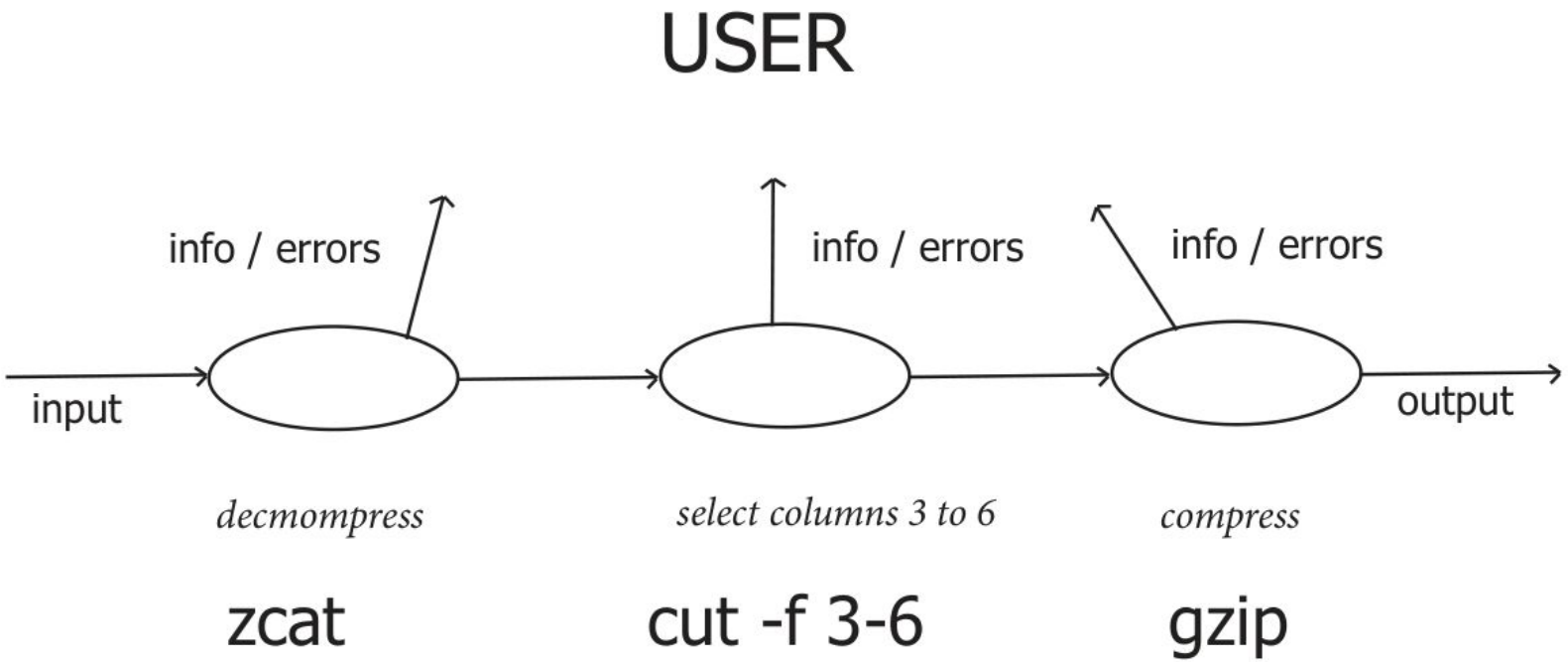
- works with **text**
- reads data line by line
- outputs data line by line
- does one simple operation

Flexibility: programs can be chained



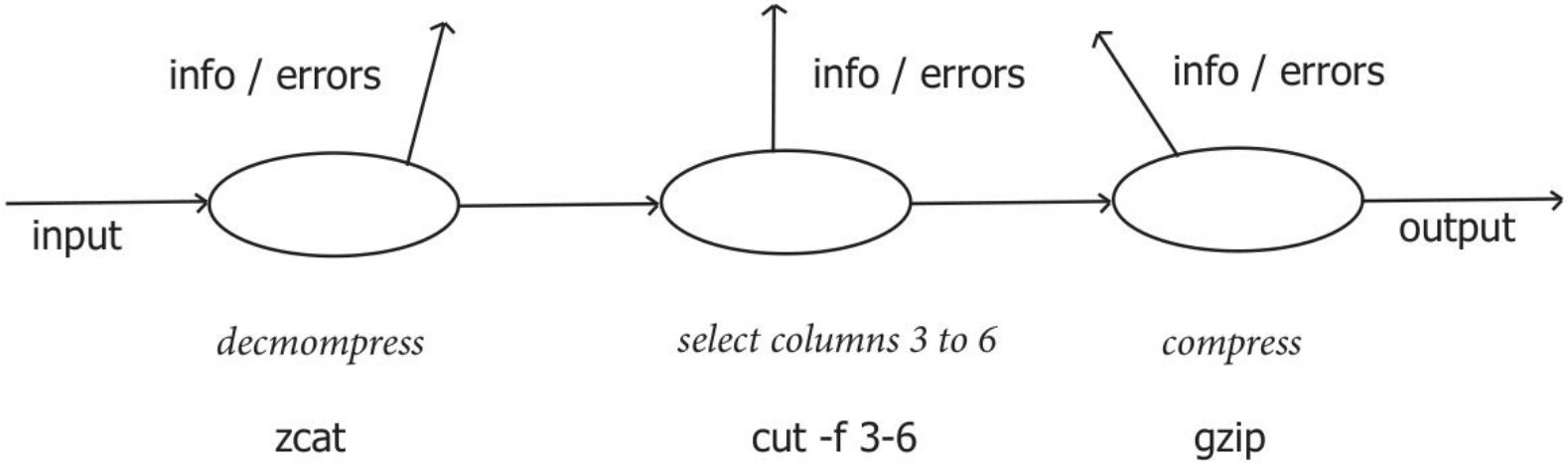
Conciseness:

"with just 30 characters you can do almost anything"



Automation:

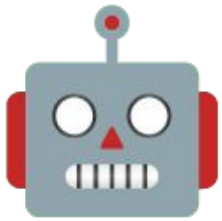
every operation can be stored as a text "recipe"



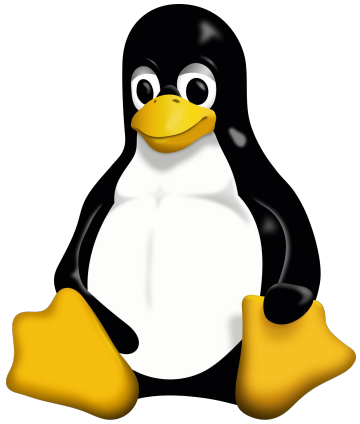
<bigtable.gz zcat | cut -f 3-6 | gzip >bigtable-3-6.gz

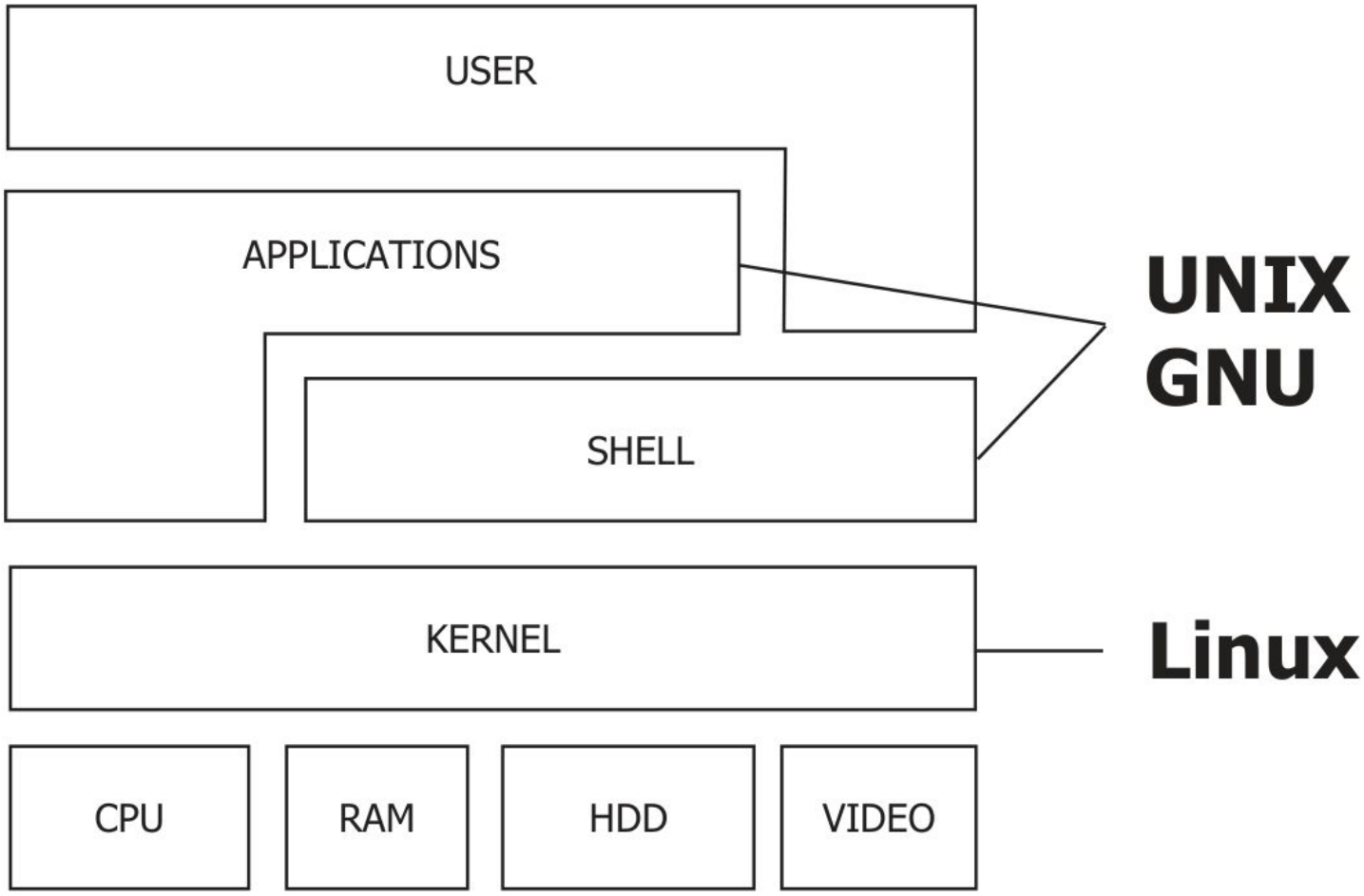
Why is it so good for genomics?

What about the robots?



And what is Linux then?




USER
APPLICATIONS
SHELL
UNIX
GNU
KERNEL
Linux
CPU
RAM
HDD
VIDEO