The Art of Installation

It begins…

Astronomy 98/198, Spring 2019

Goals

Today, we will:

- *install* **python** through a distribution called "**Anaconda**," as well as useful associated **science packages**
- *install* **git**, a version control system for tracking changes
- *cover* very basic **terminal commands**
- *download* the **Astronomy 98/198 git repository**

Mac OS (1/4)

- Open “Terminal.” (Applications ⇒ Utilities ⇒ Terminal)
- First, install brew (if you haven’t yet). Homebrew is a program that allows you to easily install other software on OSX:

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

- Type “brew --version” and confirm that you don’t run into an error
- Install wget

```
brew install wget
```

Mac OS (2/4)

- Download Anaconda installation script (if you don’t already have Anaconda)

```
wget -O install_anaconda.sh https://repo.continuum.io/archive/Anaconda3-2018.12-MacOSX-x86_64.sh
```

- Install Anaconda

```
bash install_anaconda.sh
```

- Restart your terminal (not just closing the website, but quitting the application from the bar) and reopen
- Type “conda --version” and confirm that you don’t run into an error

Mac OS (3/4)

- Run the following commands to create a new conda environment—each conda environment has its own package version, allowing easy switching between different versions of python (e.g., 3.6 and 2.7)

```
conda create --name pydecal python=3.6
source activate pydecal
conda install -n pydecal jupyter pandas numpy matplotlib
pip install okpy
```

- From now on, you can switch to the pydecal env with “activate pydecal”, and switch back to the default env with “deactivate”.

Mac OS (4/4)

- Use brew to install the latest version of git by running:

```
brew install git
```

Check that you have the latest version with "git --version"; it should be 2.5.0 or higher. You may also remove the install_anaconda.sh script from your computer, as it's quite large.

You're done.

Windows (1/3)

- Download the Python 3.6 installer for Windows (download the one that matches your computer):

32-bit: https://repo.continuum.io/archive/Anaconda3-2018.12-Windows-x86.exe

64-bit: https://repo.continuum.io/archive/Anaconda3-2018.12-Windows-x86_64.exe

- Leave all options as default, and make sure both “add to path” and “register” are checked
- Proceed with installation
- Confirm installation by opening “Anaconda Prompt”

Windows (2/3)

- Run the following commands to create a new conda environment—each conda environment has its own package version, allowing easy switching between different versions of python (e.g., 3.6 and 2.7)

```
conda create --name pydecal python=3.6
activate pydecal
conda install -n pydecal jupyter pandas numpy matplotlib
pip install okpy
```

- From now on, you can switch to the pydecal env with “activate pydecal”, and switch back to the default env with “deactivate”.

Windows (3/3)

- You might already have git installed. Type “git” into Anaconda Prompt. If that doesn’t throw an error, you can skip these steps.
- If you don’t have git installed, type the following into Anaconda Prompt:

```
conda install -c anaconda git -y
```

- Verify that you have git installed using “git –version.”

You're done.

Linux (1/4)

Note: these instructions assume that you have apt-get (Ubuntu and Debian). For other Linux distributions, substitute the available package manager.

You likely know this already if you run Linux, but just in case: your terminal program allows you to type commands to control your computer. On Linus, you can open the Terminal by going to the Applications menu and clicking “Terminal.”

Linux (2/4)

- Install wget:

```
sudo apt-get install wget
```

- Download the Anaconda installation script:

```
wget -O install_anaconda.sh https://repo.continuum.io/archive/Anaconda3-2018.12-Linux-x86_64.sh
```

- If you have a 32-bit operating system, use this command instead:

```
wget -O install_anaconda.sh https://repo.continuum.io/archive/Anaconda3-2018.12-Linux-x86.sh
```

- Install Anaconda

```
wget -O install_anaconda.sh https://repo.continuum.io/archive/Anaconda3-2018.12-Linux-x86.sh
```

- Ensure that the installation worked by running “conda --version”

Linux (3/4)

- Run the following commands to create a new conda environment—each conda environment has its own package version, allowing easy switching between different versions of python (e.g., 3.6 and 2.7)

```
conda create --name pydecal python=3.6
source activate pydecal
conda install -n pydecal jupyter pandas numpy matplotlib
pip install okpy
```

- From now on, you can switch to the pydecal env with “activate pydecal”, and switch back to the default env with “deactivate”.

Linux (4/4)

- Now install the latest version of git by running the following:

```
sudo add-apt-repository ppa:git-core/ppa
sudo apt-get update
sudo apt-get install git
```

Check that you have the latest version with "git --version"; it should be 2.5.0 or higher. You may also remove the install_anaconda.sh script from your computer, as it's quite large.

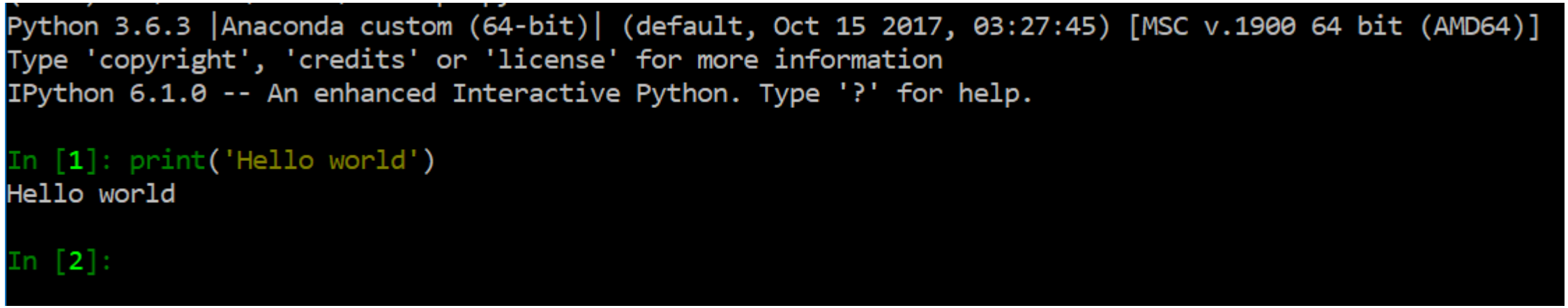
You're done.

Running Python

- Type “ipython” into terminal to start IPython, which is a python implementation which allows terminal commands and broader control

Running Python

- Type “ipython” into terminal to start IPython, which is a python implementation which allows terminal commands and broader control



Cloning our Git Repository

- Make an account at github.com (we will use this to manage Github classroom in the future)
- Open terminal and navigate to a directory that you want the repository to be
- Type the following:

```
git clone https://github.com/astro98sp19/astro98sp19.git
```

- From now on, if you navigate into the repo and type “git pull”, the repo will be updated

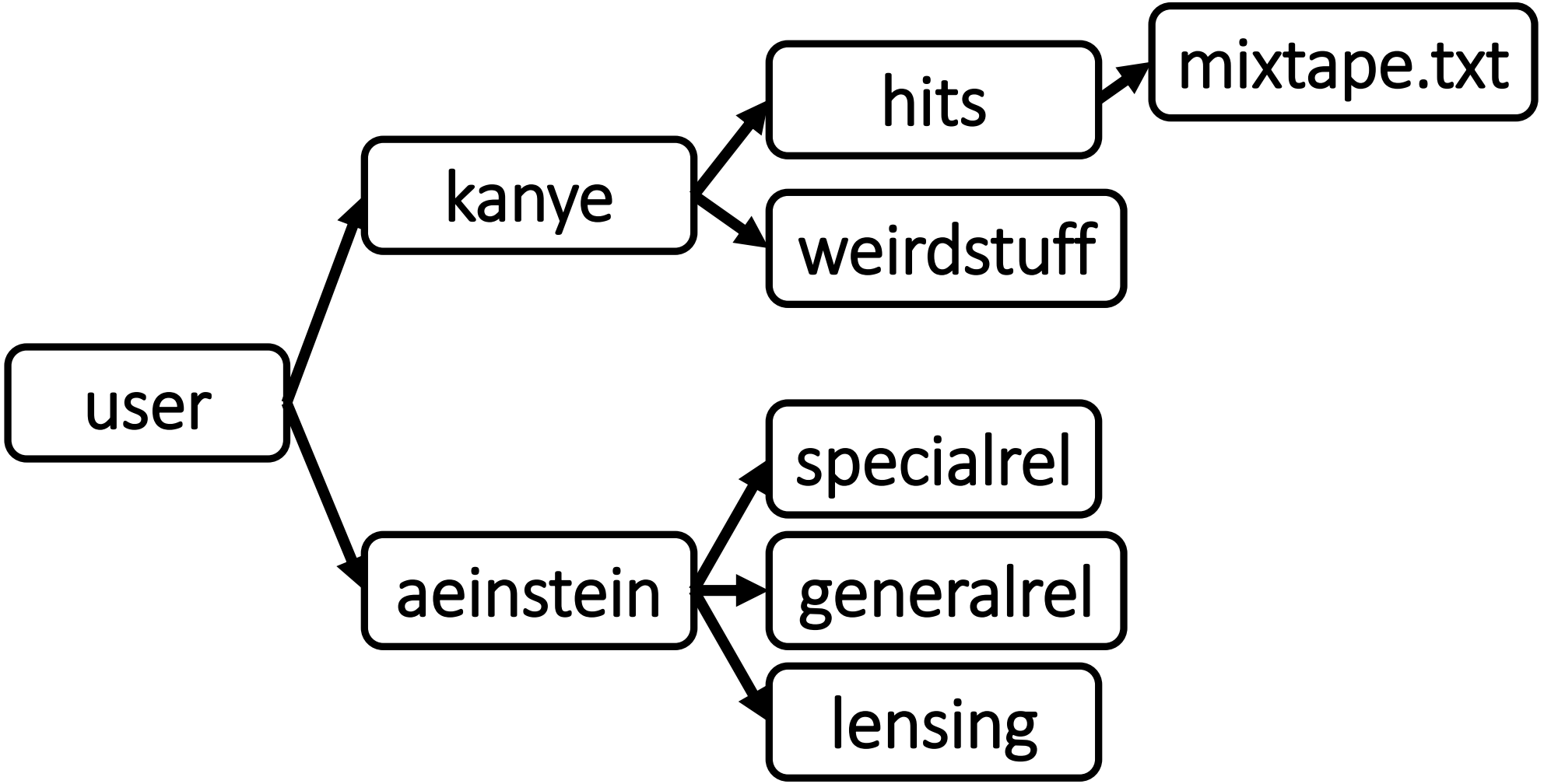
Terminal Commands: The Gist

- The command line/terminal allows you to navigate your directory structure and perform tasks using a text interface.
- Commands for Windows and Mac/Linux are slightly different, but same general idea
- Your computer is structured into “directories” (you can think of them as folders).

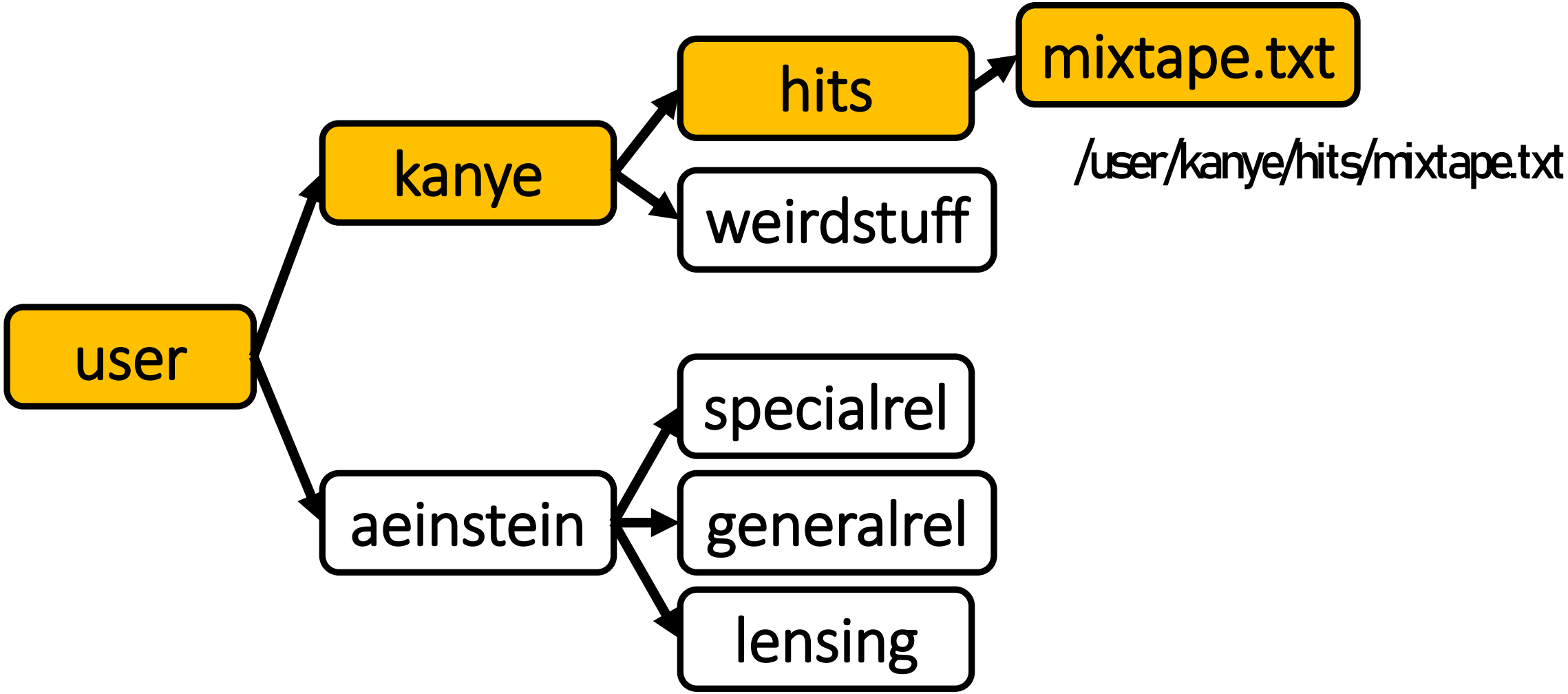
Terminal Commands: Directories

- Much like the commonly used term, directories are “addresses” where files can be located
- You can think of these as folders, with folders being nested inside other folders
- On the command line, you’re always in some current “working directory.” For Mac/Linux, type “**pwd**” to view this. For Windows, type “**cd**” (**note:** “cd” on Mac/Linux does something different)
- To see what’s in your current directory, type “**ls**” on Mac/Linux or “**dir**” on Windows.

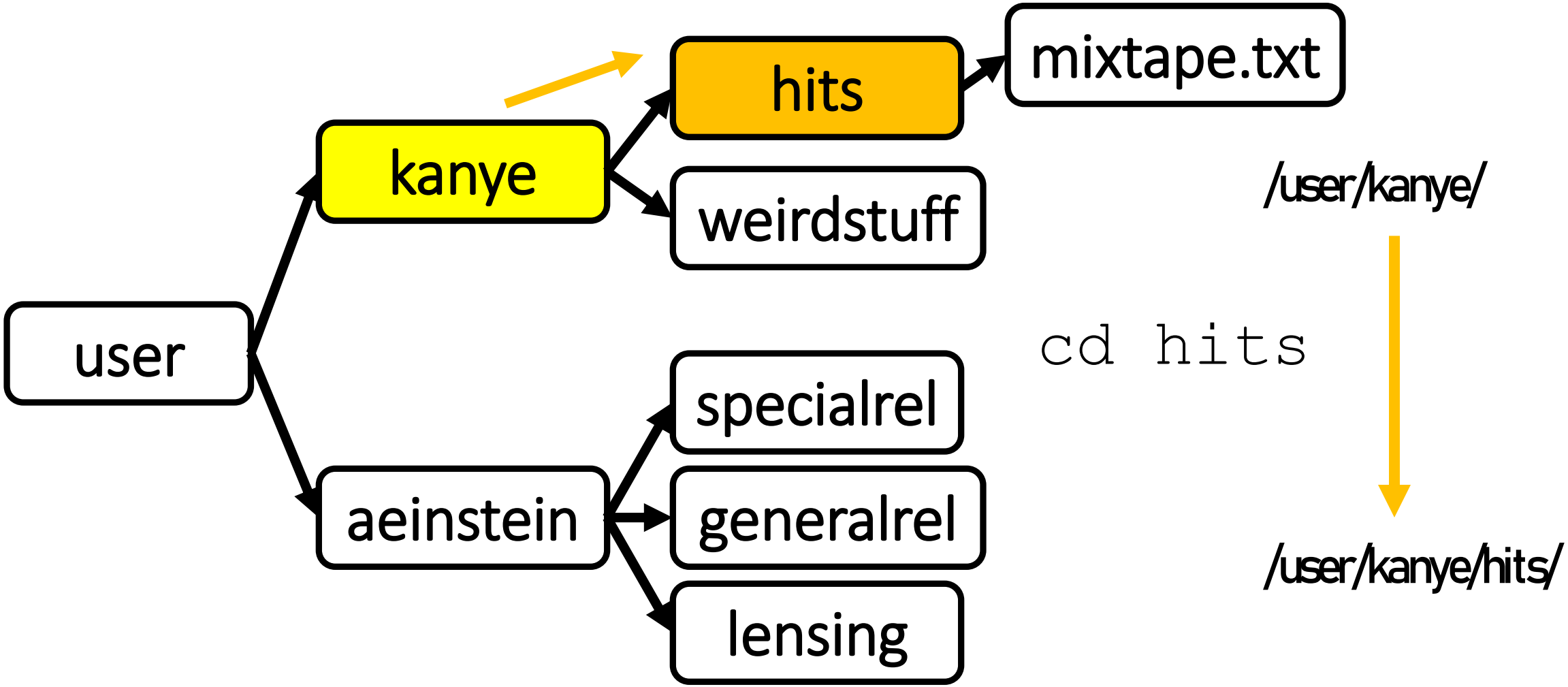
Terminal Commands: Example Directory Structure



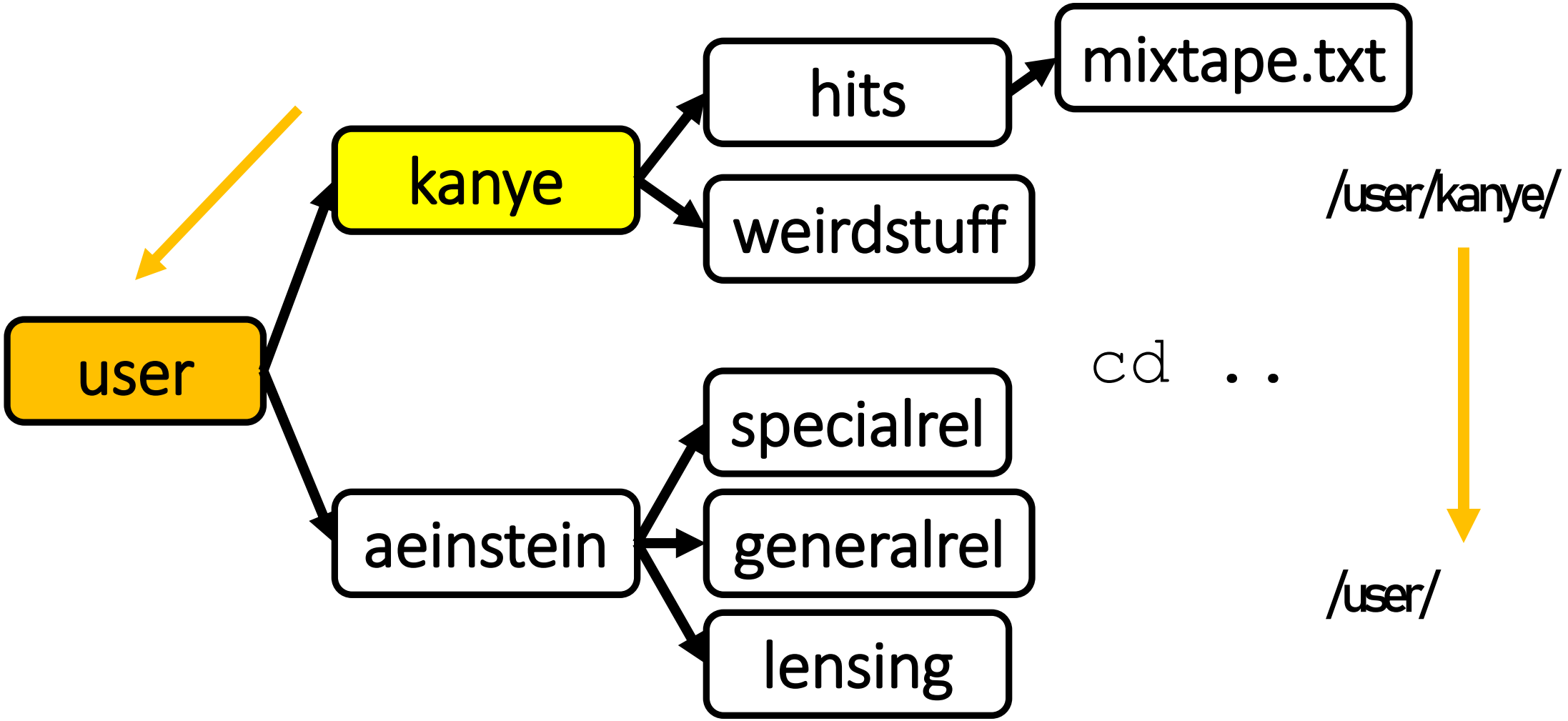
Terminal Commands: Example Directory Structure



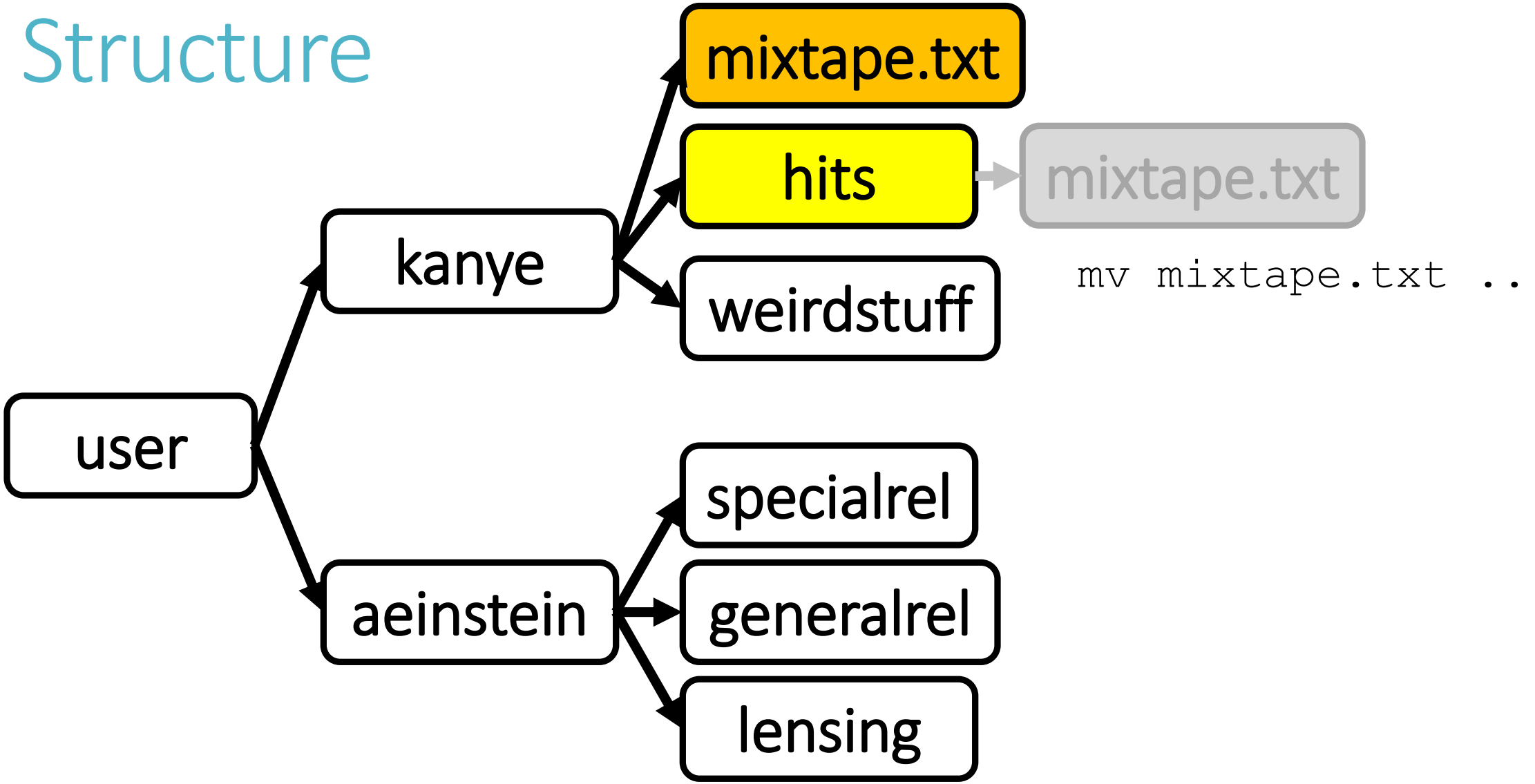
Terminal Commands: Example Directory Structure



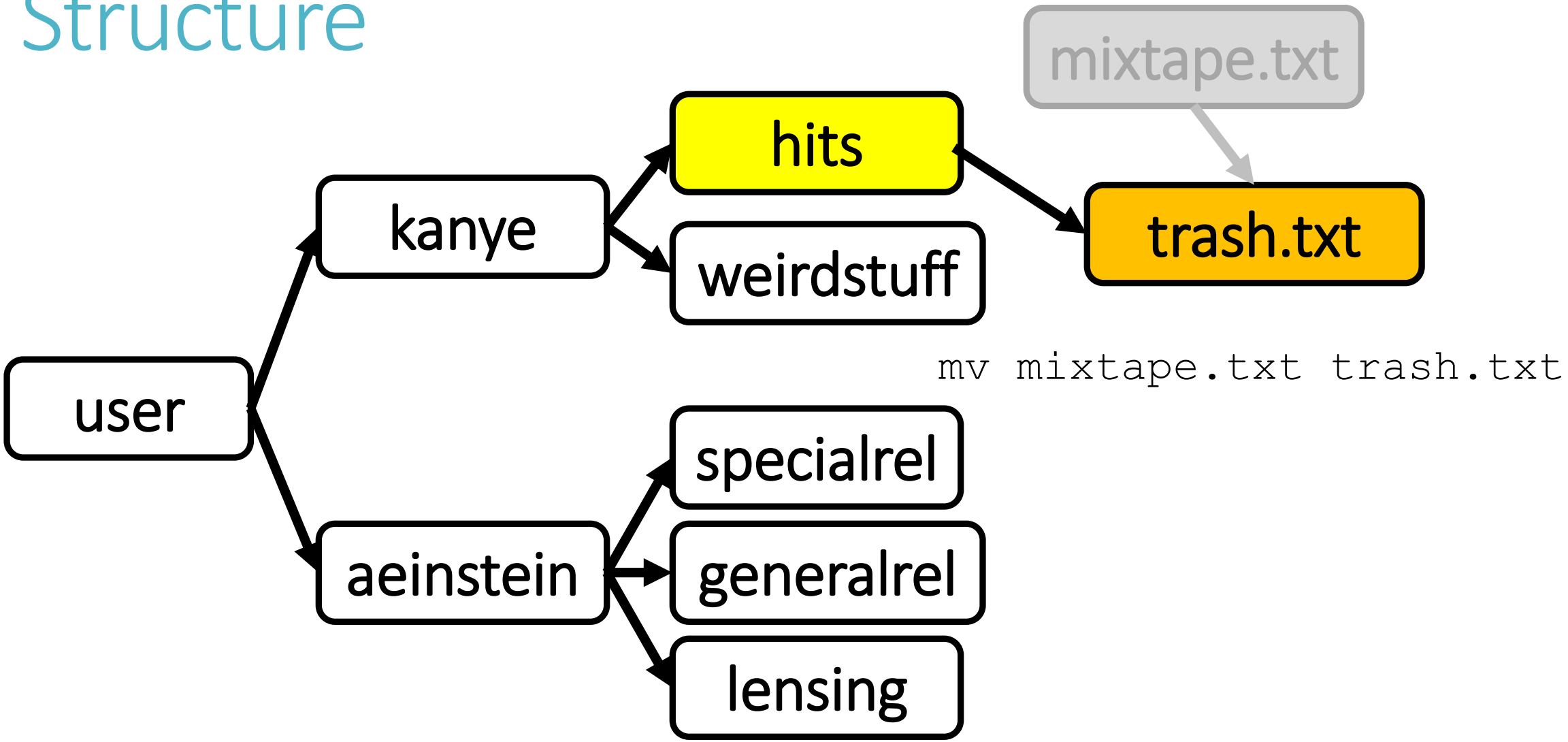
Terminal Commands: Example Directory Structure



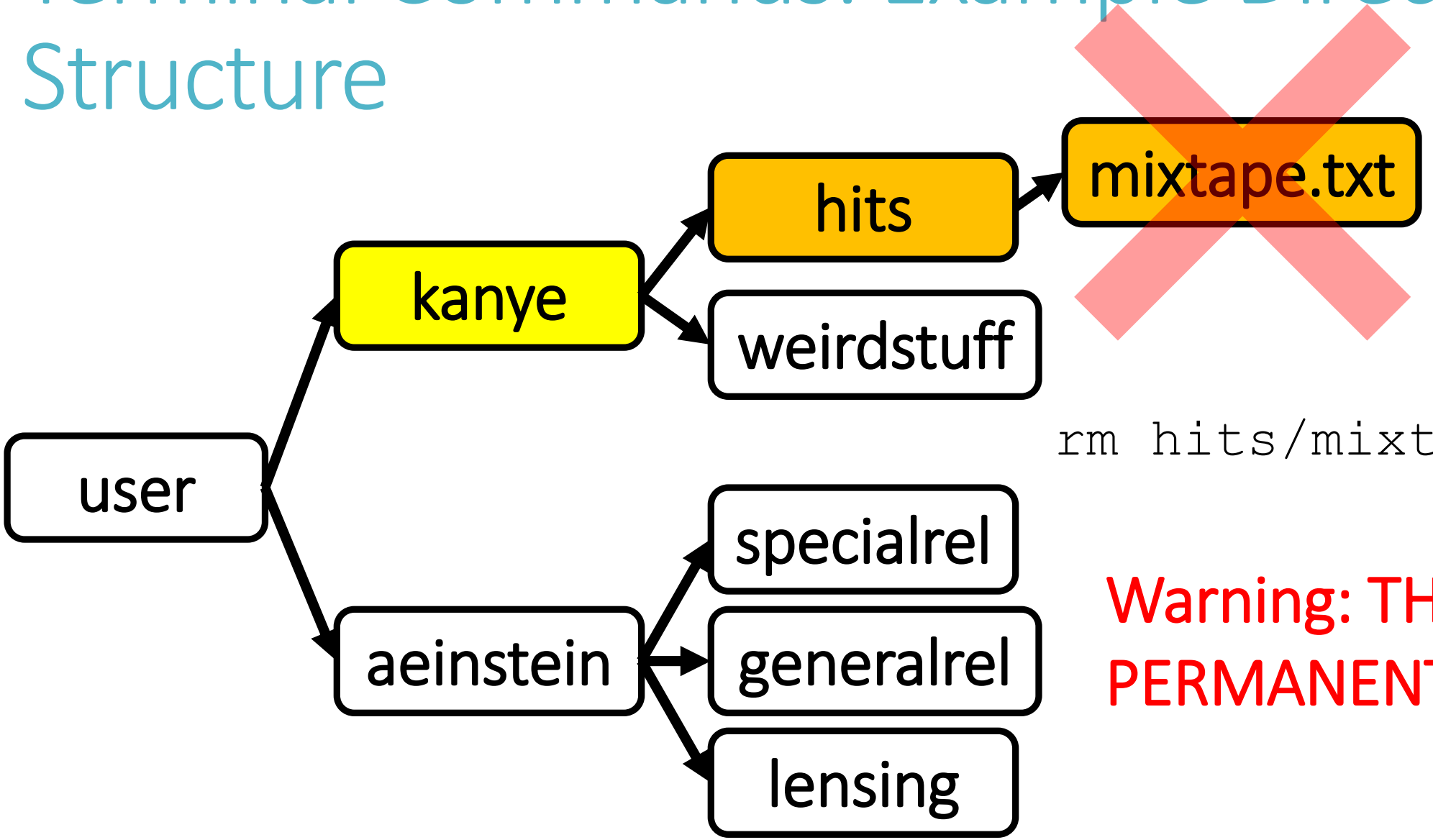
Terminal Commands: Example Directory Structure



Terminal Commands: Example Directory Structure



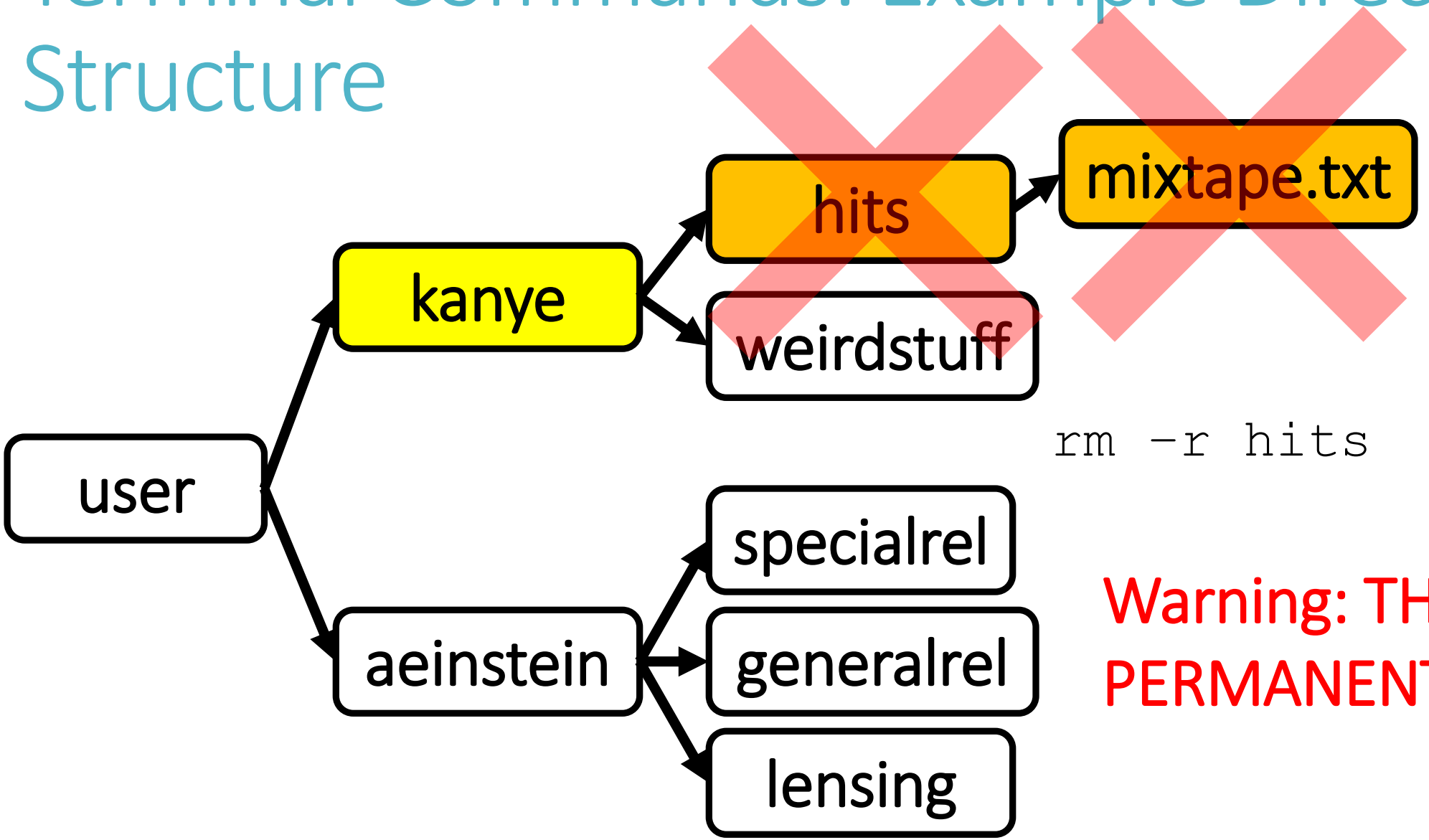
Terminal Commands: Example Directory Structure




```
rm hits/mixtape.txt
```

Warning: THIS IS PERMANENT

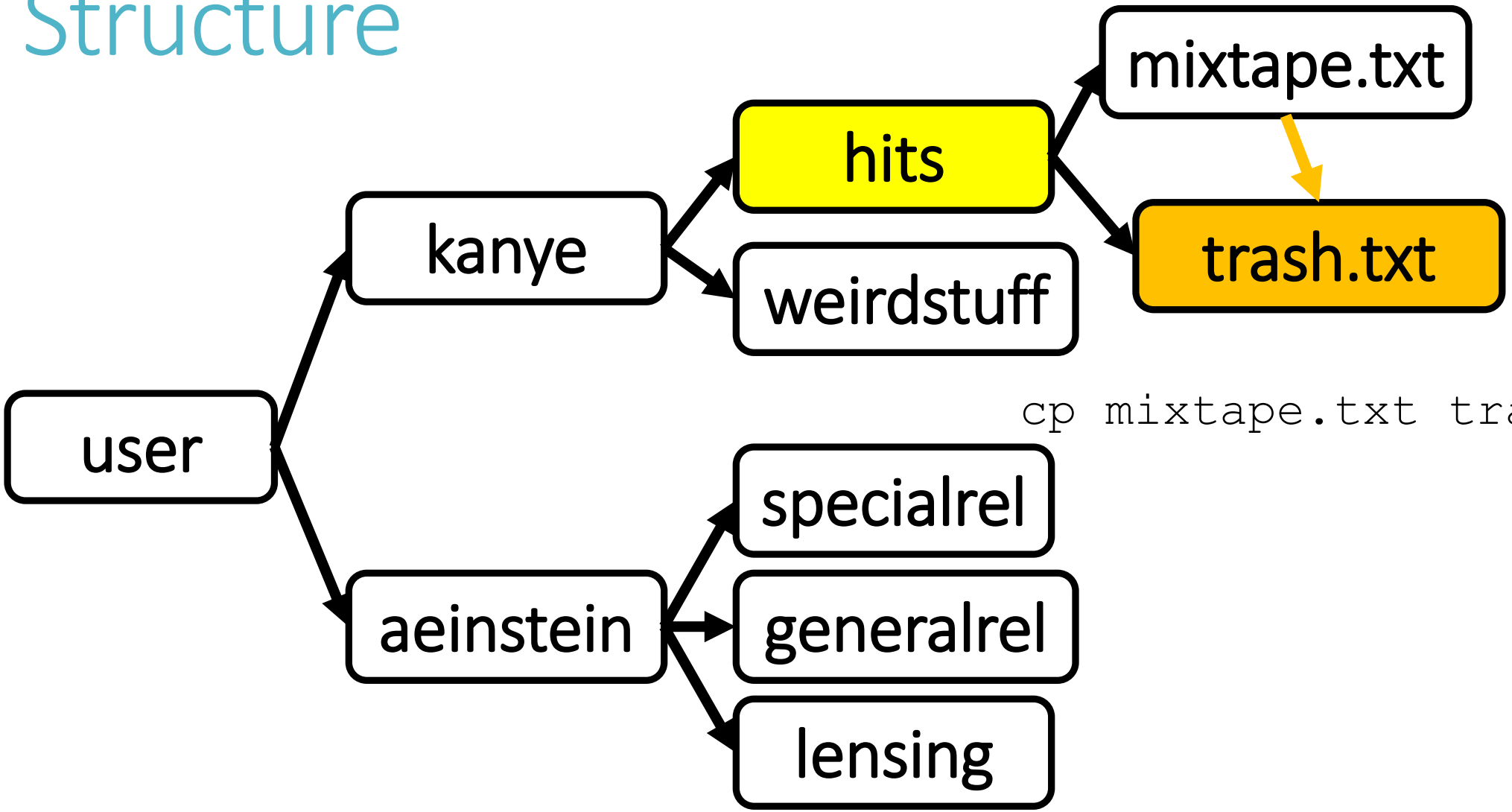
Terminal Commands: Example Directory Structure




```
rm -r hits
```

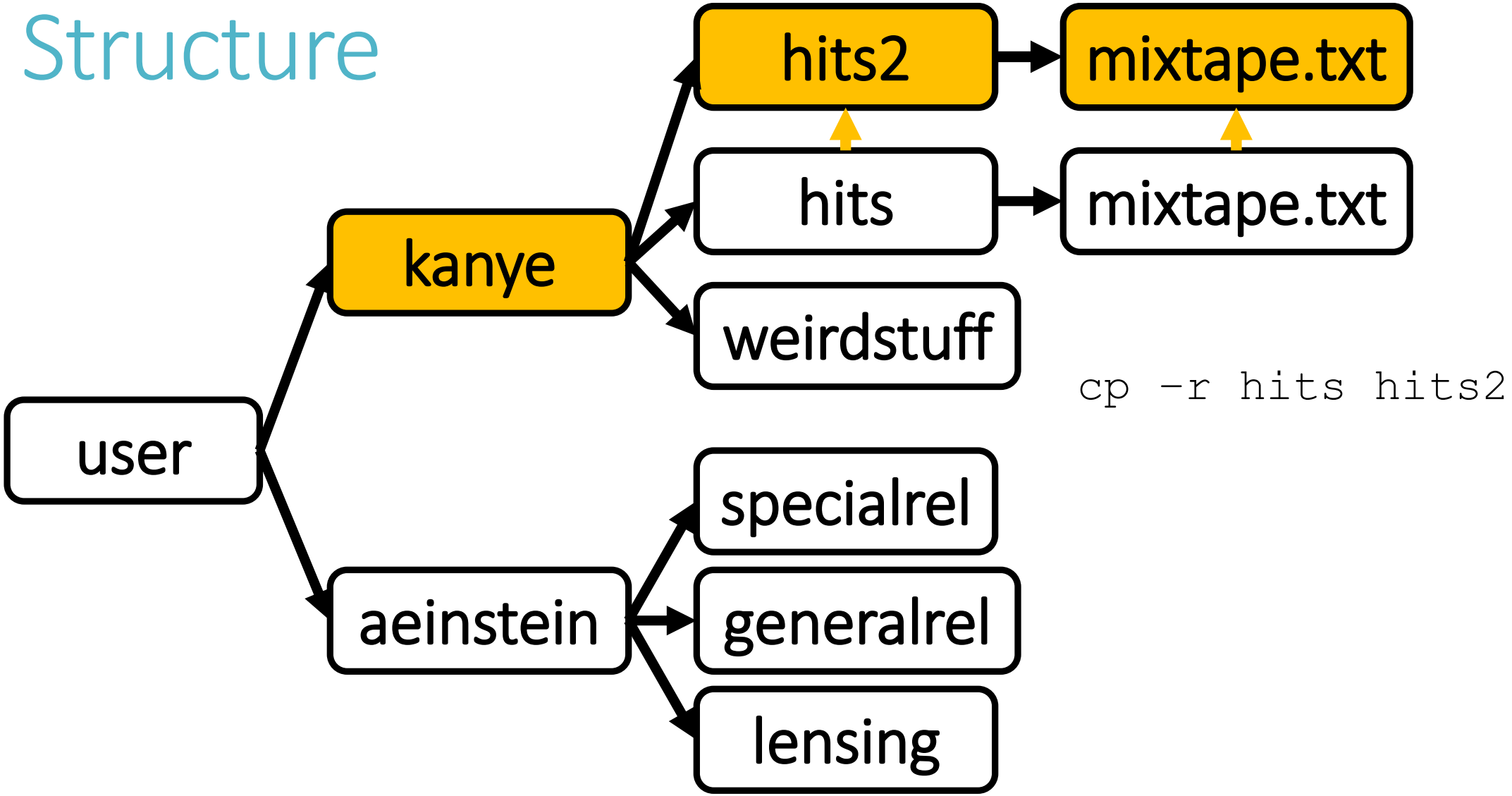
Warning: THIS IS PERMANENT

Terminal Commands: Example Directory Structure




```
cp mixtape.txt trash.txt
```

Terminal Commands: Example Directory Structure




```
cp -r hits hits2
```