ScaDS.AI
DRESDEN LEIPZIG
CENTER FOR SCALABLE DATA ANALYTICS AND ARTIFICIAL INTELLIGENCE

Introduction to Makefiles

Jan Frenzel

Dresden, March 4, 2025

GEFÖRDERT VOM
Bundesministerium für Bildung und Forschung

SACHSEN
Diese Maßnahme wird gefördert durch die Bundesregierung aufgrund eines Beschlusses des Deutschen Bundestages. Diese Maßnahme wird mitfinanziert durch Steuermittel auf der Grundlage des von den Abgeordneten des Sächsischen Landtags beschlossenen Haushaltes.

Scientist's life

- Write software
- Run applications
- Do experiments
- Collect data
- Analyze data

Run scripts, don't type commands! Automation!

Script?

Scripts come with their own challenges!

- Multiple scripts?
  - Order of invocations?
  - Dependencies?
  - Documentation?
- Workflow systems
  - How to express steps?
  - for small/large projects?
- Single script?
  - Parameters?
  - Environment variables?
  - Output to terminal or to file?
- Further considerations
  - Error handling?
  - Skip parts that already have been done?
  - How to transform scripts to automated tests/deployments?
  - Add scripts to repositories?

Makefiles to the rescue!

Makefile?

A makefile is a workflow description containing rules.
A rule consists of:

- Goals/Targets: The things you want to have/build
- Dependencies (optional): The things that are required to reach your goal/target
- Recipe (optional): Steps to do or commands to invoke.

General syntax:

```
1 # comments start with #
2 target: dependency1 dependency2
3   ./recipe
```

> **⚠ Warning**
>
> Important: Indent recipe with tabs, not spaces!

Makefile?

Invocation:

```
1 make # first target in Makefile is used
```

or

```
1 make target
```

What make does:

- Complain if a dependency does not exist or cannot be built
- Check if (sub-)target exists or needs to be created
- Rerun recipe if target is older than a dependency
- Stop immediately if one command in the recipe fails

A Pizza Example

```
# to make a pizza, you require dough and salami and the step is to bake the dough
pizza: dough salami
  ./bake dough salami

# to make dough, you need salt, water and flour. You need to mix the ingredients.
dough: salt water flour
  ./mix
```

Invocation:

```
make pizza
```

If you don't like pizza, but want to have dough for something else:

```
make dough
```

A Pizza Example

Demonstration!

PDF Example

```
# to make a dissertation pdf, you need a dissertation.tex and images without transparency
dissertation.pdf: dissertation.tex image-without-transparency.png
  pdflatex dissertation.tex
  biber dissertation
  pdflatex dissertation.tex
  pdflatex dissertation.tex

# automatically convert the image when above target is made (and the image was not converted before)
image-without-transparency.png: image.png
  convert image.png -background white -alpha remove -alpha off image-without-transparency.png

# for displaying the PDF
view: dissertation.pdf
  evince dissertation.pdf

# for displaying todos
find-todos: dissertation.tex
  grep todo dissertation.tex
```

Data analysis example

```
# We assume that there is already some script to generate a plot
plot.png: data.csv
  ./generate-plot.sh data.csv $@

# We assume that there is already some script to generate the data
data.csv: program.py
  python3 program.py

# for displaying the image
display: plot.png
  eog $<
```

- `$@` stands for the target
- `$<` stands for the first dependency

HPC script submission

```
# We don't need to remember the invocation parameters of the job script
submit-variant1:
  sbatch -J variant1 job.sbatch blub 47 11

submit-variant2:
  sbatch -J variant2 job.sbatch blub 08 15
```

Makefile?

- Defines an entry point
- Defines dependencies between goals and subgoals
- Reruns only necessary steps to achieve goals/build targets
- Is flexible (Can invoke further scripts if needed)
- Acts as a replacement for long documentation and changes while you develop
- Can often be easily integrated into automatic testing and deployment

Example for a short README:

Building

Assuming Docker and `make` is installed, you can use `make` to build the application.

Running

If you have built the Docker image, you can run it using `make run`. Please see the `Makefile` if you want to know the endpoint.

Testing

Assuming `curl` is installed, a simple test example can be started using `make test`.

Thank you for your attention!

This presentation and example are also on
`https://gitlab.hrz.tu-chemnitz.de/scads_ai_pages/intro-to-makefiles`