Introduction to Bash scripting language

Prelude

Alessandro Sciarra

Z02 – Software Development Center

17 February 2025

GOETHE
UNIVERSITÄT
FRANKFURT AM MAIN

AxelKrypton

CRC-TR 211
Strong-interaction matter
under extreme conditions

Lecture format

{17..21} February 2025

Morning: 09:15 to 12:30   { Lecture with slides }
Afternoon: 14:15 to 17:30   { Live programming: Exercise sheet – I am here to help, use me }

Monday, Tuesday and Friday: FIAS, room 0|200
Wednesday and Thursday: GSC, room 0|08

Dr. Alessandro Sciarra

sciarra@itp.uni-frankfurt.de

Who am I?

> **Ancient popular wisdom** ─────
>
> «I am not saying to be Superman, I am just saying nobody has ever seen me and Superman together in the same room.»

- Member of the Software Development Center
  - → Z02 project of the CRC-TR211 collaboration
- A used-to-be "lattice practitioner"
- A dreamer who flatters himself that
  - physicists will one day properly use the IT tools they use
  - IT people will teach IT to physicists and work next to them
  - physicists will be able to focus on physics
- Not really a Bash guru, but rather a self-thought Bash enthusiast

The mind setup of this course

The mind setup of this course

1 **Relaxed atmosphere!**
Interactive lecture: ask, ask, ask!

The mind setup of this course

Remember:
This is not the full story!

2 The amount of information given in this lecture is **huge**, I am aware of it

3 You will not receive all the details (I tried to pass on the important information, though)
↳ be active, ask, use manuals, go beyond this lecture!

1 **Relaxed atmosphere!**
Interactive lecture: ask, ask, ask!

The mind setup of this course

Remember:
This is not the full story!

2. The amount of information given in this lecture is **huge**, I am aware of it
3. You will not receive all the details (I tried to pass on the important information, though)
   ↳ be active, ask, use manuals, go beyond this lecture!
4. Often it is hard (very hard) to clarify all details of and exceptions to rules
   ↳ experience and practice will fill the gap
5. From rules/practical examples to abstract ideas/good practices

1. **Relaxed atmosphere!**
   Interactive lecture: ask, ask, ask!

The mind setup of this course

> **Remember:**
> This is not the full story!

2 The amount of information given in this lecture is **huge**, I am aware of it

3 You will not receive all the details (I tried to pass on the important information, though)
  ↳ be active, ask, use manuals, go beyond this lecture!

4 Often it is hard (very hard) to clarify all details of and exceptions to rules
  ↳ experience and practice will fill the gap

5 From rules/practical examples to abstract ideas/good practices

6 100% Bash, no focus on portability or differences among
various shell languages { I am not at that level }
  ↳ sh, ksh, csh, tcsh, dash, zsh, fish, ...

1 **Relaxed atmosphere!**
Interactive lecture: ask, ask, ask!

What is 🔗 Bash ?

🔗 Bash logo

BASH
THE BOURNE-AGAIN SHELL

- Scripting/command language
- Unix shell ⟷ command processor
- First release in 1989, current version v5.3-beta (Dec 2024) → 🔗 download Bash

> **GNU Bash**
>
> Bash is the GNU Project's shell. Bash is the Bourne Again SHell. Bash is an sh-compatible shell that incorporates useful features from the Korn shell (**ksh**) and C shell (**csh**). [...] It offers functional improvements over **sh** for both programming and interactive use. In addition, most **sh** scripts can be run by Bash without modification.

What is 🔗 Bash ?

🔗 Bash logo

BASH
THE BOURNE-AGAIN SHELL

- Scripting/command language
- Unix shell ↔ command processor
- First release in 1989, current version v5.3-beta (Dec 2024) → 🔗 download Bash

> **GNU Bash**
>
> Bash is the GNU Project's shell. Bash is the Bourne Again SHell. Bash is an sh-compatible shell that incorporates useful features from the Korn shell (**ksh**) and C shell (**csh**). [...] It offers functional improvements over **sh** for both programming and interactive use. In addition, most **sh** scripts can be run by Bash without modification.

- A scripting language which may get hard to read → you need some "code sense"
- The effort required to a newbie is quite large → you need to practice!
- ...I am here to help you fill up this gap!

Why should I know Bash?

- It is probably the shell you use* (Bash is everywhere)
- Knowing Bash a bit more allows for better problem handling
  - ↳ e.g. delegate to Bash tasks that maybe you currently do in C/C++ or Python
- It is a scripting language very close to the (Unix) operative system
- Opportunity to make your daily work-flow more comfortable

* On macOS from Catalina (10.15) on the default shell was changed from Bash to Zsh for licensing reasons. Check out 🔗here how to switch back to Bash.

Why should I know Bash?

- It is probably the shell you use* (Bash is everywhere)
- Knowing Bash a bit more allows for better problem handling
  - ↪ e.g. delegate to Bash tasks that maybe you currently do in C/C++ or Python
- It is a scripting language very close to the (Unix) operative system
- Opportunity to make your daily work-flow more comfortable

> However, 🔗 Bash weaknesses are known:
>
> - Floating point math
> - Lack of data structures
> - Try/catch and Exception handling: forget them!
> - Functions: a tricky world!
> - ...
>
> Knowing another scripting language (e.g. Python) helps!

* On macOS from Catalina (10.15) on the default shell was changed from Bash to Zsh for licensing reasons. Check out 🔗 here how to switch back to Bash.

What to expect from the lecture?

> **My point of view as physicist**
>
> The way you program today is better than it was yesterday and probably worse than it will be tomorrow! Bash has some kind of subjective corners where you will have to choose your way. Have a good reason to do things the way you do them. It will happen that you will change mind and consequently your approach to the same problem. Again and over again, it is normal. Do not hesitate in doing so. To change mind and approach means to evolve and to become a better programmer.

- This is especially important as physicist
- Knowing all details and tweaks of Bash is probably not your goal
  ↳ nor is it probably needed in your daily work
- Having an overview is, instead, very important (e.g. to decide when to use Bash)
- As soon as something smells odd, it is important to be able to have a detailed look

$$\sum \int (\ldots) = \int \sum (\ldots) \text{ until something starts looking odd}$$

References for future use

About Bash world:

🔗 Bash official manual

🔗 List of Bash builtins with descriptions

🔗 Notable changes in released Bash versions { More extensive list 🔗 also available }

🔗 A nice Bash cheatsheet

🔗 Google Shell Style Guide

🔗 A (sadly not maintained) Bash library { Clone from Github to use it, reference website is unfortunately down }

🔗 An online help to get relevant manual pages of a given command

Didactical material:

🔗 Bash Beginners Guide { A nice, from-zero tutorial }

🔗 Shell scripting tutorial { Nice tutorial, but not only Bash }

🔗 Bash Guide { Very precise guide, not completely from zero, though – AKA Greg's Wiki }

🔗 All material of this lecture

Material used to prepare this lecture

Main reference for preparing this course

My GitHub profile

What will we explore?

What will we explore?

A slide about notation

Slides notation

- Slides are (too) full of text for a later reading!

- Many examples to immediately show you how rules apply

- In Bash code, syntax highlighted to help recognise the elements

- Prompt commands are always preceded by a dollar sign

```
$ ls
```

{ When the $ is absent, the code refers to a script or it is pseudo-code }
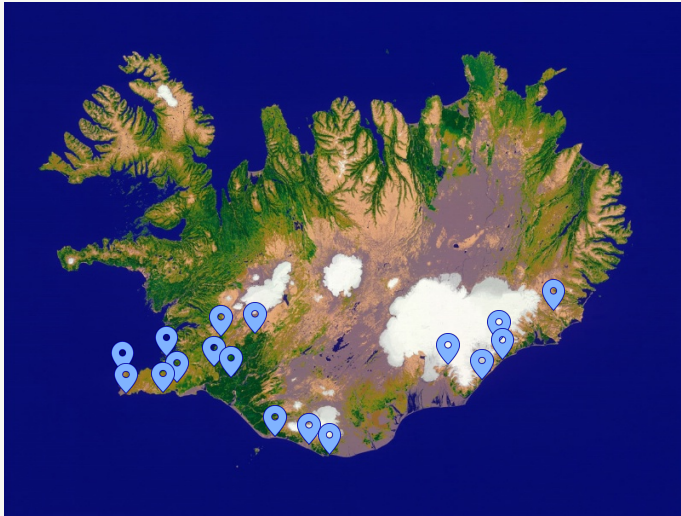
Exercise colour notation

- Review the morning lecture

- Focusing on and playing with a Bash feature

- Inspirational for real life

- Technical

Clickable link
to clickable ToC

This note usually provides further (advanced) information. Links here are worth the read.

Bonus track

Bonus track: Iceland
The magic of an incredible island

A short but intense journey
A few more recommendations

- I am aware that you will receive a lot (really a lot) of information
- Do not hesitate to ask!
- Put effort into the exercises
- The more active you are the more information will stay in your memory for your daily work

What do I expect?

A short but intense journey
A few more recommendations

- I am aware that you will receive a lot (really a lot) of information
- Do not hesitate to ask!
- Put effort into the exercises
- The more active you are the more information will stay in your memory for your daily work

What do I expect?

- You will also learn general coding principles, which are independent of the language
  ↳ Check out 🔗«Clean code» and «Clean testing» talks!
- You use the **Q&A** functionality if you do not want to orally ask
- You enjoy the journey! { ...at least discovering Iceland... }
- You give me feedback at the end of the lecture!

A short but intense journey
A few more recommendations

- I am aware that you will receive a l   ) of information
- Do not hesitate to
- Put effort into th   es
- The more active you   re in   ill st   memory for your daily work

What do I expect?

- You will also learn general codin   re independent of the language
  - ↳ Check out 🔗«Clean code»
- You use the **Q&A** functional   ally ask
- You enjoy the journey!
- You give me feedback at

Are you ready?