BVS1 · Winter Semester 2026/2027 · TH Köln · ICCT

Handout — Ubuntu Shell — Command-Line Cheat Sheet

Except where otherwise noted, © 2026 Stavros Kousidis · CC BY-SA 4.0

Revision: 2026-10-16

This sheet contains the shell operations you need most often in BVS1. Commands are entered in the Ubuntu terminal inside the `bvs1` VM unless an instruction explicitly says otherwise.

Know where you are

```
pwd
ls
ls -l
ls -la
```

- `pwd` — print the current directory;
- `ls` — list files;
- `ls -l` — detailed listing;
- `ls -la` — include hidden files such as `.git`.

Important path names:

Path	Meaning
`.`	current directory
`..`	parent directory
`~`	your home directory
`/`	filesystem root

Navigate with:

```
cd ~/teaching-bvs1-code
cd ..
cd ~
cd -
```

`cd -` switches back to the previous directory.

Let the shell complete names for you

Use **Tab completion** instead of typing long paths and filenames manually.

```
cd ~/tea<Tab>
```

If the prefix identifies one name, Bash completes it.

If several names match, press **Tab Tab** to display the available choices, type a little more, and press **Tab** again.

This reduces typing and avoids many filename mistakes.

Create, copy, rename, move

```
mkdir work
mkdir -p build/manual
touch notes.txt
cp main.c main.c.backup
cp -r source/ source-copy/
mv old-name.txt new-name.txt
mv file.txt directory/
```

Remember:

- `touch notes.txt` creates an empty file if `notes.txt` does not exist; if it already exists, `touch` updates its timestamps without deleting its contents;
- `cp` copies;

- `mv` moves **or renames**;
- `mkdir -p` also creates missing parent directories.

Before overwriting something important, inspect the source and destination with `ls -l`.

Inspect files without editing them

```
cat main.c
less main.c
head main.c
tail main.c
file build/manual/main.o
tree
```

Use `less` for longer text files:

Action in `less`	Key
Next / previous page	**Space** / **b**
Search	**/text**, then **Enter**
Next match	**n**
Quit	**q**

Search file contents with:

```
grep -n "main" main.c
grep -R "uart_puts" .
```

Find files by name:

```
find . -name '*.c'
```

Quotes matter: they prevent the shell from expanding `*.c` before `find` receives it.

Shell shortcuts worth learning

Action	Key
Complete command/path	**Tab**
Show completion choices	**Tab Tab**
Previous / next command	**Up** / **Down**
Search command history	**Ctrl-R**
Cancel the current command	**Ctrl-C**
Clear the screen	**Ctrl-L**
End input / leave a shell	**Ctrl-D**

Show recent commands with:

```
history
```

When a command fails, **read the error message before trying another command**.

Combine commands and output

Run the second command only if the first succeeds:

```
make && make run
```

Send the output of one command into another:

```
riscv64-unknown-elf-nm build/manual/main.o | less
```

Redirect output:

```
command > output.txt
command >> output.txt
```

- `>` **replaces** the file;
- `>>` appends to it.

Be as careful with `>` as with other destructive operations: an existing file can be overwritten.

Paths, spaces, and wildcards

If a path contains spaces, quote it:

```
cd "My Files"
```

Common wildcards:

Pattern	Meaning
`*`	any sequence of characters
`?`	one character
`*.c`	all names ending in `.c`

The shell expands wildcards **before** running the command. Always think about what a wildcard matches before using it with a command that modifies or removes files.

Danger zone — `rm` removes files

> ⚠ There is no trash basket for normal shell rm
>
> `rm` normally removes files **immediately**. It does not move them to the desktop Trash and there is no normal undo command.
>
> ```
> rm file.txt
> ```
>
> For directories:
>
> ```
> rm -r directory/
> ```
>
> The options mean:
>
> - `-r` — remove recursively, including directory contents;
> - `-f` — force removal and suppress many prompts/errors.
>
> Combined:
>
> ```
> rm -rf directory/
> ```
>
> is therefore powerful and potentially destructive.
> **Use `rm -rf` only when you understand exactly which path will be removed. Do not copy and paste it blindly.**

Before removing a directory, especially recursively:

```
pwd
ls -ld directory/
ls directory/
```

Check that you are in the expected place and that the target is really the directory you intend to remove.

For extra confirmation, use:

```
rm -i file.txt
rm -ri directory/
```

The interactive forms ask before removing.

Permissions and programs

Check whether a command exists:

```
command -v qemu-system-riscv64
```

Make a script executable:

```
chmod +x script.sh
```

Get help:

```
man rm
rm --help
man bash
```

Press **q** to leave a manual page.

Use `sudo` only when an instruction genuinely requires administrator privileges. Do not build or edit your BVS1 repository as root.

A good BVS1 command-line routine

```
pwd
 ↓
ls
 ↓
edit
 ↓
git diff
 ↓
make
 ↓
run / inspect
 ↓
explain what happened
```

When something goes wrong, first establish **where you are**, **what files exist**, and **which command failed**. Do not repair several layers at once.