TD1 · Shell commands

NUnix · L3 MIAGE · University of Rennes

Elisabeth Bouriah and Louis Ledoux 2026–2027

Exercise 1: filename patterns

List the entries in `/usr/bin` whose names:

1. Start with `m`.
2. Start with `m` and have exactly three characters.
3. Start with `m` and have at least three characters.
4. Start with `m` and have an extension (a suffix after a dot).

Exercise 2: redirections

Consider this directory tree. The current directory is `user2`.

```
practice/
├── user1/
│   ├── f1
│   ├── f2
│   └── f3
└── user2/
    └── f4
```

Using `cat` and redirections:

1. Create `f5` containing the line `5 2 3 -1`.
2. Create `f6` containing `f5` followed by `f4`.
3. Append `user1`'s `f1` to `f6`.
4. Copy `user1`'s `f2` into a new file `f7` in `user2`.

Exercise 3: a script with three filenames

1. Write a shell script `execution`, called with three filenames: `n1 n2 n3`. It must perform these steps:

Input	Program	Output
File `n1`	`p1`	File `n1.r1`
File `n2`	`p1`	File `n2.r1`
`n2.r1` followed by `n1.r1`	`p2`	File `n3`

   The programs `p1` and `p2` read standard input and write standard output. Assume they are executable and available in the current directory.

2. Using the tree from Exercise 2, what happens if `user1` runs the following command from their directory? Identify the input files and the location of each output file.

   ```
   ./execution f1 ../user2/f4 f5
   ```

3. Write an interactive version that asks for `n1`, `n2` and `n3` instead of receiving them as command-line arguments.

Exercise 4: command-line arguments

Write a shell script `argument` that displays its arguments one at a time:

```
$ ./argument one two three
3 arguments:
 1 - one
 2 - two
 3 - three
```

Use `expr` to update the counter. This command evaluates arithmetic and logical expressions and prints the result; for example, `expr 1 + 2` prints `3`.

Exercise 5: searching for a pattern

1. Write `mysearch`, called as `./mysearch pattern filename`. It prints either `filename has the pattern` or `filename does not have the pattern`.
2. Write `mysearch2 filename`. It repeatedly asks for a pattern to search for in the file, until the user enters `end`.
3. Write `mysearch3 f1 ... fn`. For each pattern entered by the user, search all the files supplied as arguments. Stop when the user enters `end`.

You can redirect unwanted output to `/dev/null`.

Exercise 6: the find command

Explain the effect of each command. Read them before executing them:

```
find ./ -name tp2 -exec {} < f_in > {}.out \;
find /var/log -exec grep "www.univ-rennes1.fr" {} \;
find . -name "rc.conf" -exec chmod o+r {} \;
find /usr/local -type f -name "*.html"
```