

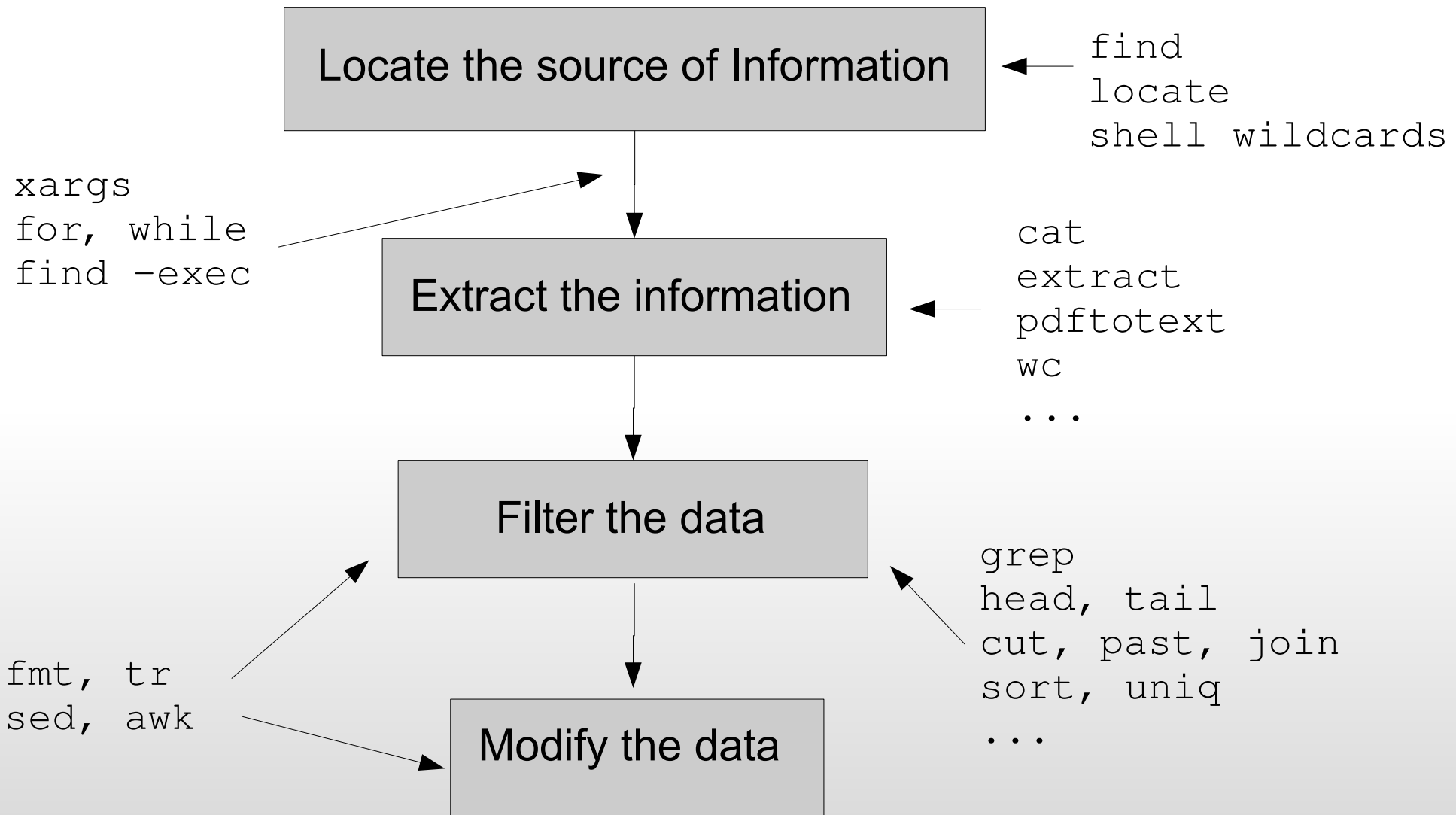
Software Development Linux ToolSet (part C)

Davide Balzarotti

Eurecom – Sophia Antipolis, France

```
> cat test | awk '/^[A-Z]/ {print $NF}'
```

```
> cat test | sed -r 's:([a-zA-Z0-9]+) \1:\1:g'
```



The Right Tool For the Right Job

Find one or more file	<code>find</code>
Find files that contains some text	<code>find grep</code>
Number of lines, words, bytes	<code>wc</code>
Print certain lines (by number)	<code>sed</code>
Print certain lines (by content)	<code>grep</code>
Extract patterns	<code>grep -o</code>
Remove lines that match a pattern	<code>grep -v</code>
Remove certain characters from all lines	<code>tr</code>
Remove/substitute a string/pattern	<code>sed</code>
Convert to lower/upper case	<code>tr</code>
Group lines (w/o frequency)	<code>sort uniq -c</code>
Intersection, union, complements of files	<code>sort uniq</code>

The Right Tool For the Right Job

Remove/keep some columns	<code>cut, awk</code>
Repeat a command for each file	<code>find -exec, find xargs</code> <code>for loop</code>
Repeat a command for each line	<code>cat while loop</code> <code>awk</code>
Repeat a command for each word	<code>fmt while, tr while, awk</code>
Insert a line	<code>sed</code>
Extract part of a file (from ... to)	<code>sed</code>
Anything that requires some math	<code>awk</code>
Working on record split on multiple lines	<code>awk</code>
Modify one or more columns	<code>awk</code>

So far...

- Something about the history
 - The origins, the Berkley era, the GNU system...
 - The people: Ken Thompson, Dennis Ritchie, Bill Joy, Richard Stallman, Linus Torvalds...
- Something about the philosophy
 - Write programs that do one thing and do it well
 - Programs are tools and like all good tools, they should be specific in function, but usable for many different purposes
 - Remember the pipe - the power is not in the tools, but in the way you combine them together
- Something about the shell and the command line tools
 - You can go a long way without writing a single line of code
 - RegEx can save you in many situation

Scripts

- Invented as a way to execute sequences of terminal commands in batch mode
- Text files containing the commands to be executed by a particular interpreter
 - Usually do not follow the compile-link-run cycle, but interpreted or compiled “just in time” (JIT)
- Distinction not so clear nowadays
- Many languages for many different purposes:
 - Shell (bash, C..) - the original, for command scripting
 - Awk – text processing
 - PHP – WWW server side
 - Javascript – WWW client side
 - Perl, Python, Ruby – general purpose languages

A Script under the Hood

```
#!/bin/awk -f  
  
BEGIN { print " - START - "  
{ print $0}  
END { print " - DONE -" }
```

A Script under the Hood

```
#!/bin/awk -f
```

```
BEGIN { print " - START - " }  
{ print $0 }  
END { print " - DONE -" }
```

If the first line starts with `#!` the shell considers the rest of the line as the name of the interpreter to be used to run the file

Two ways to run the script

```
> awk -f test.awk
```

```
> chmod a+x test.awk  
> ./test.awk
```

A World Full of Script

```
> find /bin /usr/bin/ /usr/local/bin -exec file -b {}
\; | grep -i script | sort | uniq -c

  1 a nickle script text executable
  5 a perl script text executable
  4 a python script text executable
101 a /usr/bin/perl script text executable
  1 a /usr/bin/perl -t script text executable
  1 a /usr/bin/perl -wl script text executable
 44 a /usr/bin/perl -w script text executable
  1 a /usr/bin/perl -wT script text executable
  2 a /usr/bin/ruby1.8 script text executable
  1 awk script text executable
 35 Bourne-Again shell script text executable
247 POSIX shell script text executable
 73 python script text executable
  1 setuid a /usr/bin/perl -T script text executable
```