Sed AWK and RegEx Lecture

Astronomy Laboratory at UniTS

Giuliano Taffoni – Giuseppe Murante

Regular Expressions

- Pattern to execute “search” operations
- You can search for words of a certain size, but also numbers, punctuation characters, combination of words and special characters etc.
- They are special identifiers that represent set of characters

```
date "+%m/%d/%Y" |  sed 's/\([0-9][0-9]\)\/\([0-9][0-9]\)\/\([0-9][0-9][0-9][0-9]\)/\2 \1 \3/g'
```

RegExp Identifires

- Anchors (line oriented position e.g. ^ and $)
- Character Sets (match one or more char e.g. .)
- Modifiers (how many times?)

```
"^#"
```

How to use them?

- Bash
- SED
- Python
- grep (Global Regular Expression Print)

  - PERL, Java, C → PCRE

Characters identifiers

- Anything “.”
- Any number [0-9] (\d)
- Any letter [a-z] [A-Z]

- More that one char or string (and) “|” [A-f]|[0-3]

```
[A-f]|[0-3]
```

```
echo "abc1d123" | grep [A-z]
```

Anchors

- Used to identify a position in the text

- “^“ begin of a line
- “$” End of a line

```
grep "^#" /etc/hosts
```

- \< Begin of a word (\b)
- \> end of a word (\b)

```
echo "si avvicina piano piano al tavolo nel 1996" |
sed  -n '/\bpiano\b/p'
```

Modifiers

- “*” zero or more

```
echo "si avvicina \"piano\" al tavolo nel 1996" |
sed  -r 's/i.*i/u/g'
```

- “?” Zero or one

```
echo "si avvicina \"piano\" al tavolo nel 1996" |
sed  -r 's/i.+i/u/g'
```

- “+” one or more

```
echo "si avvicina \"piano\" al tavolo nel 1996" |
sed  -r 's/i.?i/u/g'
```

- {n,m} minimum number/max number of elements

```
echo "si avvicina \"piano\" al tavolo nel 1996" | sed  -r 's/i.{1,2}i/u/g’
```

Examples

- “/^$/” empty lines
- “#.*” Bash comments
- “/^([a-z0-9_\.-]+)@([\da-z\.-]+)\.([a-z\.]{2,6})$/” Mach email

```
echo "taffoni@oats.inaf.it" |
sed -rn '/^([a-z0-9_\.-]+)@([\da-z\.-]+)\.([a-z\.]{2,6})$/p'
```

- “[\s\t]+$” “^[\s\t]+” both “^[\t]+|[\t]+$”
- 's/\(Arguments =\).*/\1 0\.45677;/g'

SED: The ultimate Stream EDitor

- Match
- Substitute
- Only on some lines
- Only on some occurrences

e.g. `sed ‘s/day/night/’ <old >new`

```
date "+%m/%d/%Y" |
sed 's/\([0-9][0-9]\)\/\([0-9][0-9]\)\/\([0-9][0-9][0-9][0-9]\)/\2 \1 \3/g'
```

Match strings

- echo "si avvicina piano al tavolo nel 1996" | sed 's/[^]*/(&)/’

- echo "si avvicina \"piano\" al tavolo nel 1996" | sed 's/[0-9][0-9]*/& &/’

- echo si avvicina piano al tavolo nel 1996 | sed 's/\([a-z]*\) \([a-z]*\).*/\2 \1/g’

- echo si avvicina piano al tavolo nel 1996 | sed -r 's/([a-z]*) ([a-z]*).*/\2 \1/g'
- echo si avvicina piano al tavolo nel 1996 | sed -r 's/(\w+) (\w+).*/\2 \1/g'

Match again

- echo si avvicina piano al tavolo nel 1996 | sed 's/[a-zA-Z]* //2’
  - Match the second occurrence

- sed 's/[^:]*//3' </etc/passwd

- echo si avvicina piano al tavolo nel 1996 | sed -e 's/[a-zA-Z]* //2' -e 's/[0-9]*//g'
  - Combining multiple commands with –e

SED as GREP

- sed -n 's/PATTERN/&/p' file
  - "-n" option will not print anything unless an explicit request to print is found. "/p" flag to the substitute command as one way to turn printing back on.

- sed -n 's/bash/&/p' </etc/passwd

Restrictions

- sed '3 s/[0-9]*//g' </etc/passwd
- sed '1,100 s/A/a/’
  - Restrict access on some lines

- find /usr | sed '/\/usr\/local\/bin/ s/\/usr\/local/\/common\/all/' | grep common
  - Restrictions on specific pattern

- sed '/^g/s/g/s/g’ ?????

Deleting and printing lines

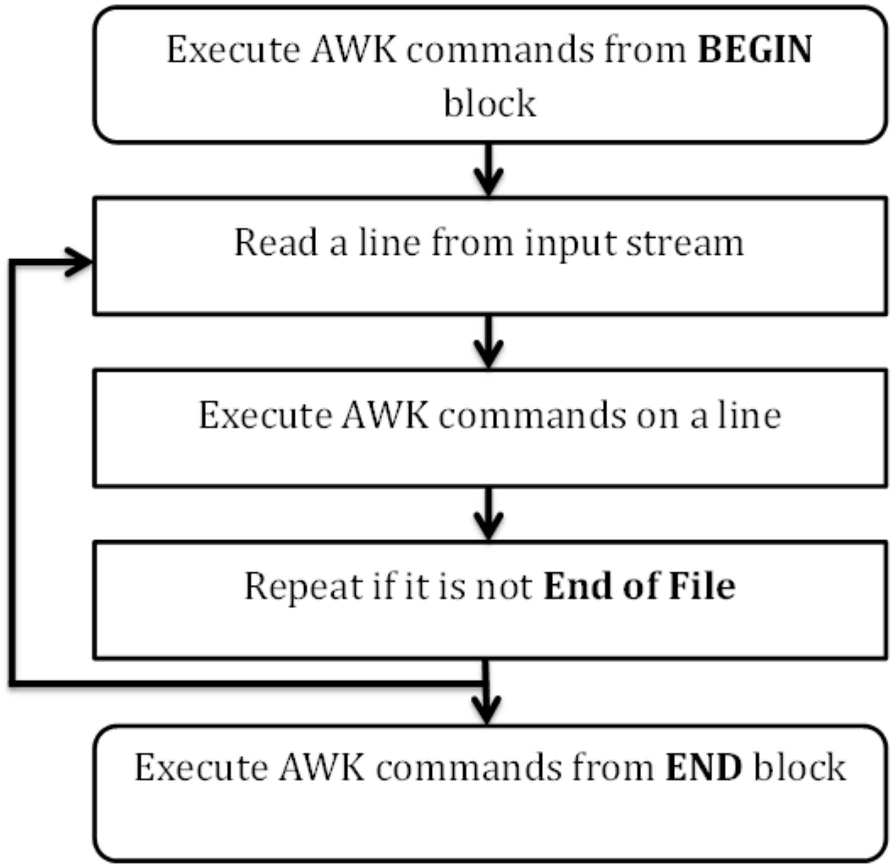
- A useful command deletes every line that matches the restriction: "d."
- `sed '/^#/ d’`

- `sed -e 's/#.*//' -e '/^$/ d’`

- `sed -n '1,10 p' <file`

AWK: a simple language

AWK is an excellent tool for processing these rows and columns, and is easier to use AWK than most conventional programming languages.


Execute AWK commands from **BEGIN** block
Read a line from input stream
Execute AWK commands on a line
Repeat if it is not **End of File**
Execute AWK commands from **END** block

A simple example

1) Amit Physics 8
2) Rahul Maths 9
3) Shyam Biology 87
4) Kedar English 85
5) Hari History 89

```
awk
'BEGIN{printf "Sr No\tName\tSub\tMarks\n"}
{print}
END{printf "END\n"}' marks.txt
```

Some examples

- awk '{print}' marks.txt
- awk -v name=Jerry 'BEGIN{printf "Name = %s\n", name}’
- awk 'BEGIN { print ENVIRON["USER"] }’
  - awk ‘{ print ENVIRON["USER"] }’

- awk '{print $1 $2}’ marks.txt
- awk '{print $NF}’ marks.txt

Other examples

- awk -F : '{print $1}’ /etc/passwd
- awk 'BEGIN { a = 10; b = a++; printf "a = %d, b = %d\n", a, b }’
- awk 'BEGIN { cnt=10; cnt += 10; print "Counter =", cnt }’
- awk 'BEGIN { a = 10; b = 20; if (a < b) print "a < b" }’
- awk 'BEGIN { str1="Hello, "; str2="World"; str3 = str1 str2; print str3 }’
- awk 'BEGIN { printf "Percentags = %d\n", 80.66 }’
- awk 'BEGIN {num = 10; if (num % 2 == 0) printf "%d is even number.\n", num }'

More Examples

- awk 'BEGIN {i = 1; while (i < 6) { printf "The square of %d is %d\n", i, i*i; ++i } }’
- awk 'BEGIN{printf "type a number\n"}{print "The square of ", $1, " is ", $1*$1;if ($1<10) print "type another number";else exit}END{print "Done"}’
- sed 's/^\(WalCoolTables_Path\).*/\1 \/u\/exanest01\/local\/CoolingTables_old/' Dianoga.par

References

http://ryanstutorials.net/regular-expressions-tutorial/

https://www.gnu.org/software/sed/manual/sed.html

https://www.gnu.org/software/gawk/manual/gawk.html

http://www.grymoire.com/Unix/