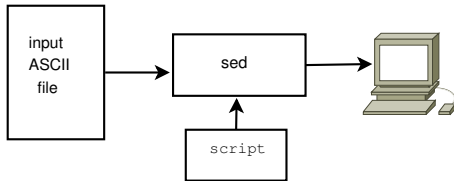


# sed and awk Programming

Spring 2025

# sed

- ▶ Character Stream Processor for ASCII files
  - not really an editor!
- ▶ Operational model: sed scans the input ASCII file on a **line-by-line** fashion and applies a **set of rules** to all lines.
- ▶ sed has three options:
  - e : script is on the command line (default case)
  - f : finds all rules that are applied in a specific (script) file.
  - n : suppresses the output



## Invoking sed

- ▶ `bash > sed -e 'address command' inputfile`
- ▶ `bash > sed -f script.sed inputfile`
- ▶ each instructions given to sed consists of an address and command.
- ▶ Sample sed-script file:

```
#This line is a comment  
2,14 s/A/B/  
30d  
40d
```

1. From lines 2 to 14 substitute the character A with B
2. Line 30 - delete it!
3. Line 40 - delete it!

sed 's/[0-9]//g'

```
gympie:~/Samples$ cat lista
john      32      london
eduardo   19      brazilia
winnie    97      cordoba
jean      21      athens
marco     7       buenosaires
filip     23      telaviv
dennis    15      brisbane
louis     31      heraclion
dimi      34      heraclion
ji        27      washington
hyseyin   33      izmir
gympie:~/Samples$
```

```
gympie:~/Samples$ cat lista | sed 's/[0-9]//g'
```

```
john      london
eduardo   brazilia
winnie    cordoba
jean      athens
marco     buenosaires
filip     telaviv
dennis    brisbane
louis     heraclion
dimi      heraclion
ji        washington
hyseyin   izmir
gympie:~/Samples$
```

## Substitution at the front and at the end of a line

```
gympie:~/Samples$ cat lista | sed 's/$/>>>/'
```

```
john      32      london>>>
eduardo   19      brazilia>>>
winnie    97      cordoba>>>
jean      21      athens>>>
marco     7       buenosaires>>>
filip     23      telaviv>>>
dennis    15      brisbane>>>
louis     31      heraclion>>>
dimi      34      heraclion>>>
ji        27      washington>>>
hyseyin   33      izmir>>>
```

```
gympie:~/Samples$ cat lista | sed 's/$/>>>/g' | \
sed 's/^/<<</g'
```

```
<<<john      32      london>>>
<<<eduardo   19      brazilia>>>
<<<winnie    97      cordoba>>>
<<<jean      21      athens>>>
<<<marco     7       buenosaires>>>
<<<filip     23      telaviv>>>
<<<dennis    15      brisbane>>>
<<<louis     31      heraclion>>>
<<<dimi      34      heraclion>>>
<<<ji        27      washington>>>
<<<hyseyin   33      izmir>>>
gympie:~/Samples$
```

## Entire-Pattern and Numbered-Buffer Substitutions

- ▶ `&` : designates the entire pattern (just matched).
- ▶ `\(` and `\)`: designate a **numbered pattern** later on identified by its respective number-id such as: `\1`, `\2`, `\3`, etc.

&  
s/-----/---&-----/

\1    \2        \3  
s/^(---)\|(------)\|(------)\|/---\1---\2-----\3---/

## Examples with Entire/Numbered-Buffers Substitutions

```
gympie:~/Samples$ cat telefona
Alex Delis          6973304567
Mike Hatzopoulos   6934400567
Thomas Sfikopulos  6945345098
Stavros Kolliopulos 6911345123
Aggelos Kiagias     6978098765
gympie:~/Samples$
```

```
gympie:~/Samples$ cat telefona | sed \
's/\([0-9]\{4\}\)\([0-9]\{2\}\)\([0-9]\{4\}\)/\1-\2-\3/'
```

```
Alex Delis          6973-30-4567
Mike Hatzopoulos   6934-40-0567
Thomas Sfikopulos  6945-34-5098
Stavros Kolliopulos 6911-34-5123
Aggelos Kiagias     6978-09-8765
gympie:~/Samples$
```

## Another Example

```
gympie:~/Samples$ cat pricelist
```

```
**This is the price list**  
  of good today  
Breakfast      10.03  
Lunch          11.45  
Dinner         7.56
```

```
gympie:~/Samples$ sed 's/[0-9]/$&/' pricelist
```

```
**This is the price list**  
  of good today  
Breakfast      $10.03  
Lunch          $11.45  
Dinner         $7.56
```

```
gympie:~/Samples$ sed 's/[0-9]/$&/3' pricelist
```

```
**This is the price list**  
  of good today  
Breakfast      10.$03  
Lunch          11.$45  
Dinner         7.5$6  
gympie:~/Samples$
```



## Local and global substitutions

```
gympie:~/Samples$ cat text2
I had a black dog, a white dog, a yellow dog and
a fine white cat and a pink cat as well as a croc.
These are my animals: dogs, cats and a croc.
```

```
gympie:~/Samples$ cat text2 | sed '1 s/dog/DOG/g'
```

```
I had a black DOG, a white DOG, a yellow DOG and
a fine white cat and a pink cat as well as a croc.
These are my animals: dogs, cats and a croc.
```

```
gympie:~/Samples$ cat text2 | sed '1 s/dog/DOG/'
```

```
I had a black DOG, a white dog, a yellow dog and
a fine white cat and a pink cat as well as a croc.
These are my animals: dogs, cats and a croc.
```

```
gympie:~/Samples$ cat text2 | sed 's/dog/DOG/g'
```

```
I had a black DOG, a white DOG, a yellow DOG and
a fine white cat and a pink cat as well as a croc.
These are my animals: DOGs, cats and a croc.
```

```
gympie:~/Samples$ cat text2 | sed '1,2 s/cat/CAT/2'
```

```
I had a black dog, a white dog, a yellow dog and
a fine white cat and a pink CAT as well as a croc.
These are my animals: dogs, cats and a croc.
gympie:~/Samples$
```



## Transforming Characters (option y)

```
gympie:~/Samples$ more text2
I had a black dog, a white dog, a yellow dog and
a fine white cat and a pink cat as well as a croc.
These are my animals: dogs, cats and a croc.
```

```
gympie:~/Samples$ cat text2 | sed 'y/abcdt/ADCBQ/'
```

```
I hAB A DlACk Bog, A whiQe Bog, A yellow Bog AnB
A fine whiQe CAQ AnB A pink CAQ As well As A CroC.
These Are my AnimAls: Bogs, CAQs AnB A CroC.
gympie:~/Samples$
```

## Additional sed Input and Output Commands

- ▶ Next (n): forces sed to read the next text line from input file.
- ▶ Append Next (N): adds the next input line to the current content of the pattern space.
- ▶ Print (p): copies the current content of the pattern space to the standard output.
- ▶ Print First Line (P): prints the content of the pattern space upto and including a newline character.
- ▶ List (l): displays “hidden” characters found in the lines of the file.
- ▶ Read (r): reads from a file
- ▶ Write (w): writes to a file

## The Next Command (n)

```
gypmie:~/Samples$ cat sedn
/^[a-z]/{
  n
  /~$/d
}
```

```
gypmie:~/Samples$ cat -n text2
 1 I had a black dog, a white dog, a yellow dog and
 2
 3 a fine white cat and a pink cat as well as a croc.
 4
 5
 6
 7 These are my animals: dogs, cats and a croc.
gypmie:~/Samples$ sed -f sedn text2
I had a black dog, a white dog, a yellow dog and

a fine white cat and a pink cat as well as a croc.

These are my animals: dogs, cats and a croc.
gypmie:~/Samples$
```

→n forces sed to read the next line from input. Before reading the next line, sed copies the current content of the pattern space to the output, deletes the current text in the pattern space, and then refills it with the next input line. After reading, it applies the script.

## Append Next (N) command

```
gympie:~/Samples$ cat text3
11111111
22222222
bbbbbbbb
cccccccv
jhds kjhj
ldjlkjds
lkdjsj44
gympie:~/Samples$
```

```
gympie:~/Samples$ more sedN
{
N
s/\n/ /
}
gympie:~/Samples$
```

```
gympie:~/Samples$ !sed
sed -f sedN text3
11111111 22222222
bbbbbbbb ccccccv
jhds kjhj ldjlkjds
lkdjsj44
```

→ While `n` clears the pattern space before inputting the next line, `append (N)` does not; it adds the next input line to the current content of the pattern space.

## A more interesting example with command N

```
gympie:~/Samples$ cat text2
I had a black dog, a white dog, a yellow dog and

a fine white cat and a pink cat as well as a croc.

These are my animals: dogs, cats and a croc.

This is a test
gympie:~/Samples$
```

```
gympie:~/Samples$ cat sednotN
/^$/ {
    $!N
    /\n$/D
}
gympie:~/Samples$
```

```
gympie:~/Samples$ sed -f sednotN text2
I had a black dog, a white dog, a yellow dog and

a fine white cat and a pink cat as well as a croc.

These are my animals: dogs, cats and a croc.

This is a test
gympie:~/Samples$
```

## Understanding the script

- What happens, should you replace D with d?
  - ▶ \$!N means “if line is not the last line”
  - ▶ \$N means “if line is the last line in the text”
  - ▶ D command: delete up to the first embedded newline in the pattern space. Start next cycle, but skip reading from the input if there is still data in the pattern space.
  - ▶ d command: delete pattern space. Start next cycle.



# The *p* command

```
gympie:~/Samples$ sed -n '2,3p' text3
```

```
22222222  
bbbbbbbb
```

```
gympie:~/Samples$ sed 'p' text3
```

```
11111111  
11111111  
22222222  
22222222  
bbbbbbbb  
bbbbbbbb  
cccccccv  
cccccccv  
jhds kjhj  
jhds kjhj  
ldjlkjds  
ldjlkjds  
lkdjsj44  
lkdjsj44  
gympie:~/Samples$
```

P command: prints content of the pattern-space upto including a newline char

```
gympie:~/Samples$ cat text4
I had a black dog, a white dog,
a yellow dog and a pink lion
    a fine white cat and
    a pink cat as well as a croc.
These are my animals:
dogs, cats and a croc.
This is a test
gympie:~/Samples$
```

```
gympie:~/Samples$ cat setprintkt
$!N
/>\n /P
D
```

```
gympie:~/Samples$ sed -f setprintkt text4
a yellow dog and a pink lion
    a fine white cat and
gympie:~/Samples$
```

## A good way to see "invisible" characters

```
gympie:~/Samples$ sed -n 'l' text4
```

```
I had a black dog, a white dog, $  
a yellow dog and a pink lion$  
\ta fine white cat and $  
\ta pink cat as well as a croc.$  
These are my animals: $  
dogs, cats and a croc.$  
This is a test$  
gympie:~/Samples$
```

## Reading files in a text with r

```
gympie:~/Samples$ cat maintext

This is blah blah blah...
and more blah blah blah blah..
and even more....
blah blah blah...
gympie:~/Samples$ cat mainheader
  THIS IS THE TEXT
gympie:~/Samples$ cat maindate

Sat Mar  6 18:17:14 EET 2010
gympie:~/Samples$
```

```
gympie:~/Samples$ cat sedread
1 r mainheader
$ r maindate
gympie:~/Samples$
```

```
gympie:~/Samples$ sed -f sedread maintext

  THIS IS THE TEXT
This is blah blah blah...
and more blah blah blah blah..
and even more....
blah blah blah...

Sat Mar  6 18:17:14 EET 2010
gympie:~/Samples$
```

## Separating lines to different files with `w` command

```
Mon 7:00 Get up!  
Tue 7:00 Get up!  
Wed 7:00 Get up!  
Thu 7:00 Get up!  
Fri 7:00 Get up!  
Mon 7:30 Get Washed  
Tue 7:30 Get Washed  
..... etc etc
```

```
gympie:~/Samples$ cat sedwrite  
/Mon/w Mon.log  
/Tue/w Tue.log  
/Wed/w Wed.log  
/Thu/w Thu.log  
/Fri/w Fri.log  
gympie:~/Samples$ sed -nf sedwrite log-events
```

```
gympie:~/Samples$ cat sedwrite  
/Mon/w Mon.log  
/Tue/w Tue.log  
/Wed/w Wed.log  
/Thu/w Thu.log  
/Fri/w Fri.log  
gympie:~/Samples$ ls *log  
Fri.log Mon.log Thu.log Tue.log Wed.log  
gympie:~/Samples$
```