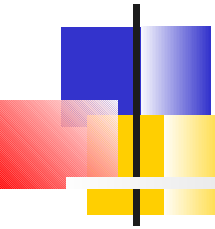
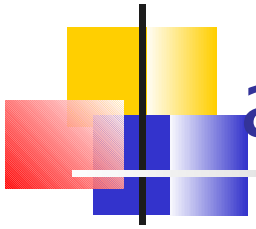


awk (powerful text parser and more)

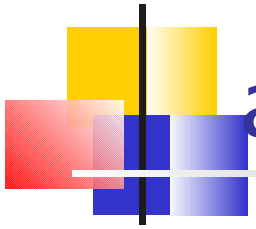
CS 2204

*Notes adapted by Alexey Onufriev from notes by Doug Bowman and other members of the VT CS faculty



awk

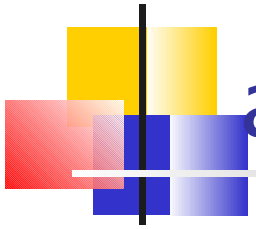
- Powerful pattern scanning and processing language
- Names after its creators Aho, Weinberger and Kernighan
- Most commands operate on entire line
- awk operates on fields within each line
- Usage:
  - `awk options [scriptfile] file(s)`
  - `Example: awk textfile > output.file`
  - Example: `awk –f awk.script textfile`



awk: Processing model

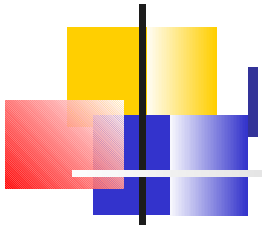
```
BEGIN {commands executed before any
  input is read}
{
Main input loop for each line of input
}
END {commands executed after all input
  is read}
```

Important: acts on each line of input text.
BEGIN and END are optional.



awk: First example

```
# Begin Processing
BEGIN {print "Print Totals"}
# Body Processing
{total = $1 + $2 + $3}
{print $1 " + " $2 " + " $3 " =
  "total}
# End Processing
END {print "End Totals"}
```



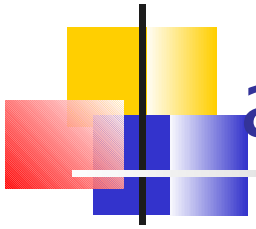
Input and output files

- Input

```
22 78 44
66 31 70
52 30 44
88 31 66
```

- Output

```
Print Totals
22 + 78 + 44 = 144
66 + 31 + 70 = 167
52 + 30 + 44 = 126
88 + 31 + 66 = 185
End Totals
```



awk: command line processing

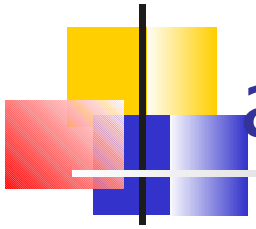
```
awk '{ if ($2 == "computers") print}' sales.dat
```

- Input

```
1 clothing 3141
1 computers 9161
1 textbooks 21312
2 clothing 3252
2 computers 12321
2 supplies 2242
2 textbooks 15462
```

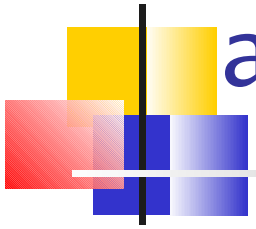
- Output

```
1 computers 9161
2 computers 12321
```



awk: Other features

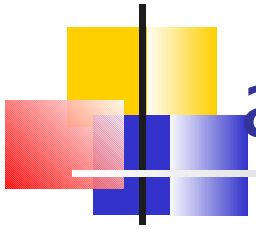
- Formatted printing using printf
- Conditional statements (if-else)
- Loops
  - for
  - while
  - do-while



awk: can use like a C-code

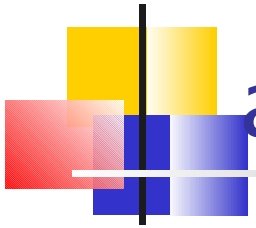
- Example: gyration.awk -- radius of a protein.
- Below are a few sample lines.
- This is just a sample file created by Dr. Alexey V. Onufriev
- ATOM 20 O THR 3 108.979 130.710 -21.735 1.00 92.2
- ATOM 21 CB THR 3 111.011 130.684 -24.119 1.00 91.
- ATOM 22 OG1 THR 3 111.240 129.529 -23.302 1.00 92

Need to compute the center (x0,y0,z0) of the protein: average of fields 6, 7 and 8. Then compute square average of (x_i - x0)^2 + ... for each atom.



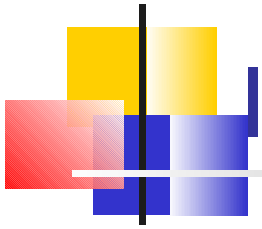
awk: Associative arrays

- Normal arrays use integers for their indices
- Associative arrays with strings as their indices
- Example: Age[“Robert”] = 56



awk: Example

```
# salesDeptLoop.awk script
BEGIN {OFS = "\t"}
{deptSales [$2] += $3}
END {for (item in deptSales)
{
print item, ":", deptSales[item]
totalSales += deptSales[item]
} # for
print "Total Sales", ":", totalSales
} # END
```



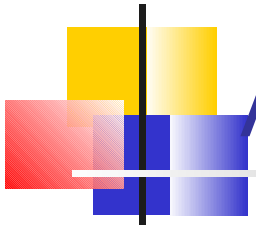
Input and output

- Input

```
1 clothing 3141
1 computers 9161
1 textbooks 21312
2 clothing 3252
2 computers 12321
2 supplies 2242
2 textbooks 15462
```

- Output

```
computers : 21482
supplies : 2242
textbooks : 36774
clothing : 6393
Total Sales : 66891
```



Another awk example

```
BEGIN {count=0; sum=0}
{if($1>=10 && $1 <= 100){
  count++
  sum+=$1
  }
else{
  print $1,"is not a double digit number"
  }
}
END{print "There are",count,"double digit
  numbers, which sum to",sum}
```