CS 101
Computer Programming and utilization

IIT BOMBAY

Dr Deepak B Phatak
Subrao Nilekani Chair Professor
Department of CSE, Kanwal Rekhi Building
IIT Bombay

Session 15, Analysis of midsem performance
Friday, September 24, 2010

Midsem marks file

1,100020002,JOSHI ADITYA B,11,1,2,3,1.5,4,1,2,2,2,0,0,0,0,0,18

2,100020006,MANDHANE AMOL B,11,2,2,3,3,4,4,2,2,2,1,5,0,0,0,30

50,100040039,ABHISHEK KUMAWAT,14,,,,,,,,,,,,,-5

A different solution

- A simple method to find the mid semester average marks for lab batches using the given data is to use a programming language called AWK
- Named after the designers at Bell Labs who invented it in 1970's
- Alfred Aho, Peter Weinberger, Brian Kernighan
- This language makes heavy use of string data type, associative arrays, and regular expressions
  - Some inadequacies led to a language called Perl

AWK language fundamentals

- AWK is a language for processing files of text.
- A file is treated as a sequence of records, and by default each line is a record.
- Each line is broken up into a sequence of fields
  - so we can think of the first word in a line as the first field, the second word as the second field, and so on.
- An AWK program is a sequence of pattern-action statements.
- AWK reads the input a line at a time, till all lines are read
  - A line is scanned for each pattern in the program, and for each pattern that matches, the associated action is executed.

How AWK handles records

Each record of our file is like

2,100020006,MANDHANE AMOL B,11,2,2,3,3,4,4,2,2,2,1,5,0,0,0,30

$1 $2 $3 $4 $19

- AWK separates out various “fields” as it reads records and assigns values to $1, $2 etc
- What do we want to do?

  Pattern: $19 < 0

  Action: Increment a count variable for absent students

  For other patterns: increment batch-counts, marktotals, …

  At END, print the accumulated results

AWK Script

```
BEGIN{ FS = ","}
$19 < 0 { absentcount++;}
$19 >=0 { count++;
          batch[$4] ++;
          totmarks += $19;
          batchcount[$4] ++;
          batchtot[$4] += $19;
        }
```

AWK Script …

```
END{for (i in batch){
    print i, batchcount[i],
          batchtot[i]/ batchcount[i];
    }
print "Total students are:  ", count + absentcount;
print " Number absent is:  ",  absentcount;
print "Class average is:  ",   totmarks/count;
}
```

AWK execution Output

```
$mawk -f analysemidsem2010v1.awk midsemmarks2010v1.txt | sort
11 13 18.1923
12 13 22.4615
13 13 21.9231
14 12 15.625
15 13 19.7692
16 12 22.25
17 11 20.9545
21 12 18.9583
22 12 23.4583
23 13 17.6538
```

Output …

74 14 13.1429

75 13 12.7308

Class average is: 16.7791

Number absent is: 19

Total students are: 569

```
/* midsemanalysisv1.cpp
   CS101 Autumn 2010-11
   Program to analyse data from midsemmarks file

   This version only finds the class average and batch averages
   Extend this to find other statistics

*/
#include <iostream>
#include <cstring>
#include <cstdlib>
using namespace std;
```

Program ...

```
int main () {
  char line[256], rollstr[8], namestr[60], batchstr[2], waitchar;
  float qmarks[20], qtotmarks[20], qaverage[20];
  char partstring[100];
  float batchtotmarks[100], batchcount[100], batchaverage[100];
  float extractedmarks, marks[15], totmarks =0, classaverage;
  int extractedbatch, nstudents=0, i, j, k, l, m, poscomma,
    poscommas[20], nchar, numcommas, startpos, endpos;
  int totabsent =0, totpresent=0;
// general initializations
  for (i=0; i<100; i++){
    batchtotmarks[i] =0; batchcount[i] = 0;
  }
```

Program …

```
//   get the first line from file
    gets(line); cout << line;
    while (line[0] != '#'){
      for (i =0; i < 20; i++) poscommas[i] = 0;
      nstudents ++;
//    Analyse the line to get important components
      nchar = strlen(line);
      for (j= 0, numcommas=0; j < nchar; j++){
        if (line[j] ==',') {poscommas[numcommas] = j; numcommas++;}
      }
```

```
    for (i=0; i <= numcommas; i++){
// ignore all the fields for the time being, except batch number and
 //total marks
      if (i==0) startpos = 0; else startpos = poscommas[i-1]+1;
      if (i==numcommas) endpos = nchar;
      else endpos = poscommas[i];
      for (k = startpos, j=0; k < endpos; k ++, j ++){
        partstring[j] = line[k];
      }
      partstring[j] = '\0';
      if (i==3) extractedbatch = atoi(partstring);
```

```
if (i == numcommas) {
        extractedmarks = atof(partstring);
        if (extractedmarks >=0){
         batchcount[extractedbatch]++;
         batchtotmarks[extractedbatch] += extractedmarks;
         totmarks += extractedmarks;
         totpresent ++; totmarks += extractedmarks;
        }
        else totabsent++;
      }
    }
 gets(line);
  }
```

```
classaverage = totmarks/totpresent;
   cout << " Total Number of students" << nstudents << endl;
   cout << "Present:  " << totpresent << ",  Absent: " << totabsent;
   cout << “     class average is: " << classaverage << endl;
   cout << "batch averages are:" << endl;
   for (i =0; i < 100; i ++){
      if (batchcount[i] !=0){
         cout << i << "  " << batchtotmarks[i]/batchcount[i] << endl;
      }
   }
  return 0;
}
```