C Assignment - Arrays

Swapnil Basak
EE11B122
IIT Madras

August 14, 2012

Problem Statement

The task is to create a C program that goes through a text file and prints the statistics of number of words with different lengths. We will consider words of length of 3 to 10 for our statistics. This program is written by indexing through strings using arrays.

Code

```
1  #include<stdio.h>
2  #include<stdlib.h>
3  #define MAXLENGTH 512
4  /*
5  Program to find out no. of words with N
6  chars breaking at ' ','.','/t','/0',' '
7  This is the Array method implementation.
8  Swapnil Basak
9  EE11B122
10 */
11
12 int main(int argc, char **argv)
13 {
14 /* Expects a filename too, check for that */
15 if(argc != 2)
16 {
17   printf("Usage ./a.out <filename>");
18   exit(1);
19 }
20 /* File pointer */
21 FILE *fp;
22 fp=fopen(argv[1], "r");
23 if(fp == NULL)
24 {
25   printf("File could not be opened");
26   exit(2);
27 }
28 char buf[MAXLENGTH];
29 int counter = 0;
30 char wordcount[11];
31 int wordlen=0;
32 int tempcount=0;
33 /* Initialize all to 0 */
34 for(counter=0;counter<11;counter++)
35 {
36   wordcount[counter]=0;
37 }
38 /* Loop till EOF */
39 while(fgets(buf, MAXLENGTH, fp))
40   {
```

```
41    printf("Parsing text line --> %s", buf);
42      for(counter=0;counter<MAXLENGTH;counter++)
43      {
44        /* Split at specified characters and check if it is in between a sentence
45        or at the beginning */
46        if(buf[counter]==' '||buf[counter]=='\t'||buf[counter]==',')
47        {
48        /* Second condition avoids double spaces and spaces after periods */
49          if(counter!=0)/*&&((counter-tempcount)>1))*/
50          {
51            wordlen=counter-tempcount;
52            wordcount[wordlen]+=1;
53            wordlen=0;
54          }
55        /* Move tempcount to present space+1 */
56        tempcount=counter+1;
57        }
58        /* Specific logic for EOF as vars need to be reset */
59        if(buf[counter]=='\0')
60        {
61          wordlen=counter-tempcount-1;
62          wordcount[wordlen]+=1;
63          wordlen=0;
64          tempcount=0;
65          /* End iteration */
66          break;
67        }
68      }
69    }
70  for(counter=1;counter<11;counter++)
71  {
72    printf(" \nThe no. of %d lettered words are %d \n", counter, wordcount[counter]);
73  }
74  fclose(fp);
75  return 1;
76  }
```

Output

```
1  ./a testfile.txt
2  Parsing text --> This is a test file with default words to see if it parses correctly
3
4  The no. of 1 lettered words are 0
5  The no. of 2 lettered words are 4
6  The no. of 3 lettered words are 1
7  The no. of 4 lettered words are 3
8  The no. of 5 lettered words are 1
9  The no. of 6 lettered words are 2
10 The no. of 7 lettered words are 1
11 The no. of 8 lettered words are 0
12 The no. of 9 lettered words are 1
13 The no. of 10 lettered words are 0
```

Algorithm

An array index ptr is declared that indexes throughout the buffer string. Two flags are used for the logic.

- A file stream is opened with specified txt file, exception is raised if not
- *buf* is initialized per line with MAXLENGTH till EOF is reached
- *counter* that travels with the array index

- *tempcount* that relates to the last position of *counter*
- *wordcount* is stored in an array with *wordcount*[n] denotes no. of words consisting of n letters.
- Finally results are displayed