

Lab 1 -- using UNIX and lex

Kay Sweebe
CS370
January 18, 2018

Document Structure

Input: A file

Output: A frequency count of alphabetic strings and the number of numbers found within the file.

Definitions:

counting array lgths of size 100

counting int count initially set to 0 [added]

Rules:

For each word found ([a-zA-Z]+), we increment the counter in the counting array at the index value equivalent to the size of the word.

For each number found ([0-9]+), we increment the counter count. [added]

If the character is any character except newline (.) or newline (\n), do nothing.

wordlengthlab1.l

```
/* This lex routine uses a counting array to match alphabetic strings
and make a frequency count.
The real item to notice is that yywrap() is called at EOF and then is run
to do what we need to do. yywrap() returns true when we have a successful
end to the program. We may want to return false (0) if we want to lexing process
to fail

Shaun Cooper
January 2015
*/
```

```

int lgths[100];
int count=0;

%%
[a-zA-Z]+      {lgths[yyvaleng]++; // this is a comment
                                   // increments the frequency count for the given size of
                                   // a word when a word is found
                                   }
[0-9]+         {count++; // the curly braces are used to make compound statements
                                   // increments the counter when a number is found
                                   }
.              |
\n             {;                // if it is any other character other than new line or new line do nothing
%%

yywrap()
{
    int i;
    printf("Length No. words\n");

    // Print the size of the words and their corresponding frequency
    for (i=1; i<100; i++) {
        if (lgths[i] > 0) {
            printf("%5d%10d\n",i,lgths[i]);
        }
    }

    // Print the number of numbers found
    printf("Number of numbers: %d \n ",count);
    return(1);
}

main()
{ yylex();

}

```

Makefile

```
all:
    # run lex on wordlengthlab1.l
    lex wordlengthlab1.l
    # compile file to get an executable, wordlength
    gcc -o wordlength lex.yy.c
```

Output

```
ksweebe@lappy29:~/Documents/CS370/Lab1> ./wordlength < /etc/passwd
Length  No. words
1       43
2       4
3       111
4       58
5       43
6       38
7       56
8       7
10      7
11      4
12      1
14      2
15      1
Number of numbers: 82
```