System Calls

Assignment 2

WAP to implement a wc(word count) command

with –l –w -c options

Assignment 2

Pre-requisites:-

- Good knowledge File operation system calls.
- Command line arguments

Objective:-

- To understand implementation of basic system call

Assignment 2

Requirements:-

1. Different ways of passing input through cmd line.

→./word_count

→./word_count –l –w

→./word_count f1.txt

→./word_count f1.txt f2.txt f3.txt

→./word_count –l f1.txt –w f2.txt f3.txt

Assignment 2

Requirements:-

2. Use getopt() function

```
int getopt(int argc, char * const argv[], const char *optstring);
```

Assignment 2

Requirements:-

2. Use getopt() function

→ int ret = getopt(argc, argv, “lwc”);

It will return alphabet(‘s) next to ‘-’ character.

Assignment 2

Requirements:-

3. Main function will open the files in a loop and call word_count function depends upon number of files passed. Print values after calling functions in main.

→int word_count (int fd, int *lines, int *words, int *bytes);

Assignment 2

Requirements:-

3. Main function will open the files in a loop and call word_count function depends upon number of files passed. Print values after calling functions in main.

```
for(i  =  0; i  <  2; i++)
{
.
.
fd = open(…);
word_count(fd, &lines, &words, &bytes);
print_values(…);
.
.
}
```

Assignment 2

Requirements:-

3. If options passed [-l –w –c] print only respective values.

```
→./word_count –l  f1.txt
  5


→./word_count  -l  f1.txt  f2.txt  -c
 5  30  f1.txt
 2  15  f2.txt
 7  45  total


→./word count  –l  -w
  ….(Cntl+D)
  1 4
```

Assignment 2

Sample Output:-

```
→./my_copy
    ...(Cntl + D)
    5  5  35

→./my_copy f1.txt f2.txt
    5 7 40  f1.txt
    3 4 10  f2.txt
    8 11 50 total
```