

Count Words in a File

Yung-Hsiang Lu

(Our) Definition of Words

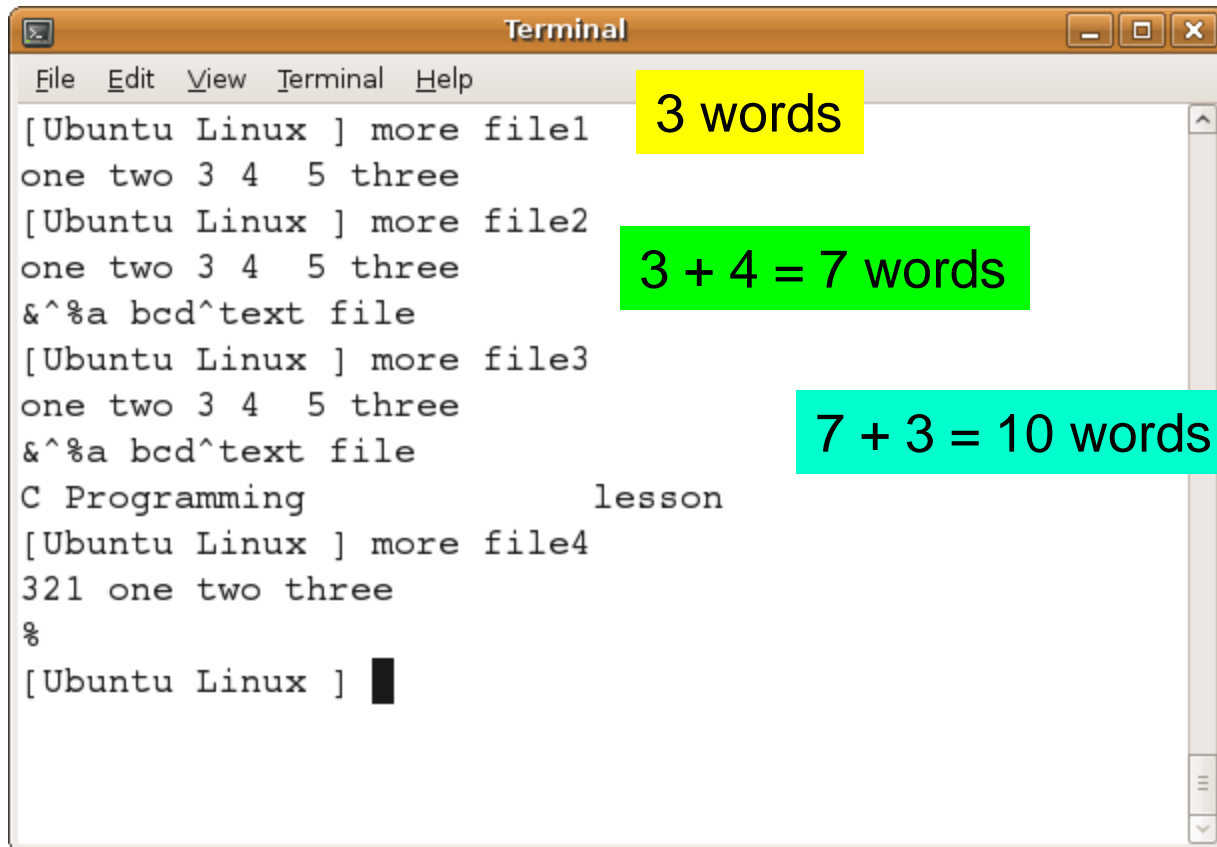
- one single character in $\{a, b, c, \dots z\}$ or $\{A, B, C, \dots Z\}$.
- If x is a word and y is a word, xy (concatenation) is a word.
 - h is a word
 - i is a word
 - “hi” is a word
- Two words are separated by at least one **non**-word character, i.e. anything not in $\{a, b, c, \dots z\} \cup \{A, B, C, \dots Z\}$.
 - “hi this is C.” contains four words: “hi”, “this”, “is”, “C”.
 - “% ^ & 7 6 u th - 0 (* & mn” contains three words: “u”, “th”, “mn”.

Regular Expression

- $[a - z]$ means one character in $\{a, b, c, \dots z\}$
- $[a - z \mid A - Z]$ means one character in $\{a, b, c, \dots z\} \cup \{A, B, C, \dots Z\}$.
- $[a - z \mid A - Z]^+$ means one or multiple characters in $\{a, b, c, \dots z\} \cup \{A, B, C, \dots Z\}$
- examples in $[a - z \mid A - Z]^+$
 - u
 - hy
 - bYTm
 - oNe

Write a Program Counting the Number of Words in a File

Input Files



A terminal window titled "Terminal" with a menu bar (File, Edit, View, Terminal, Help) and standard window controls. The terminal shows a sequence of commands and their outputs. Three colored callouts are present: a yellow one for "3 words", a green one for "3 + 4 = 7 words", and a cyan one for "7 + 3 = 10 words".

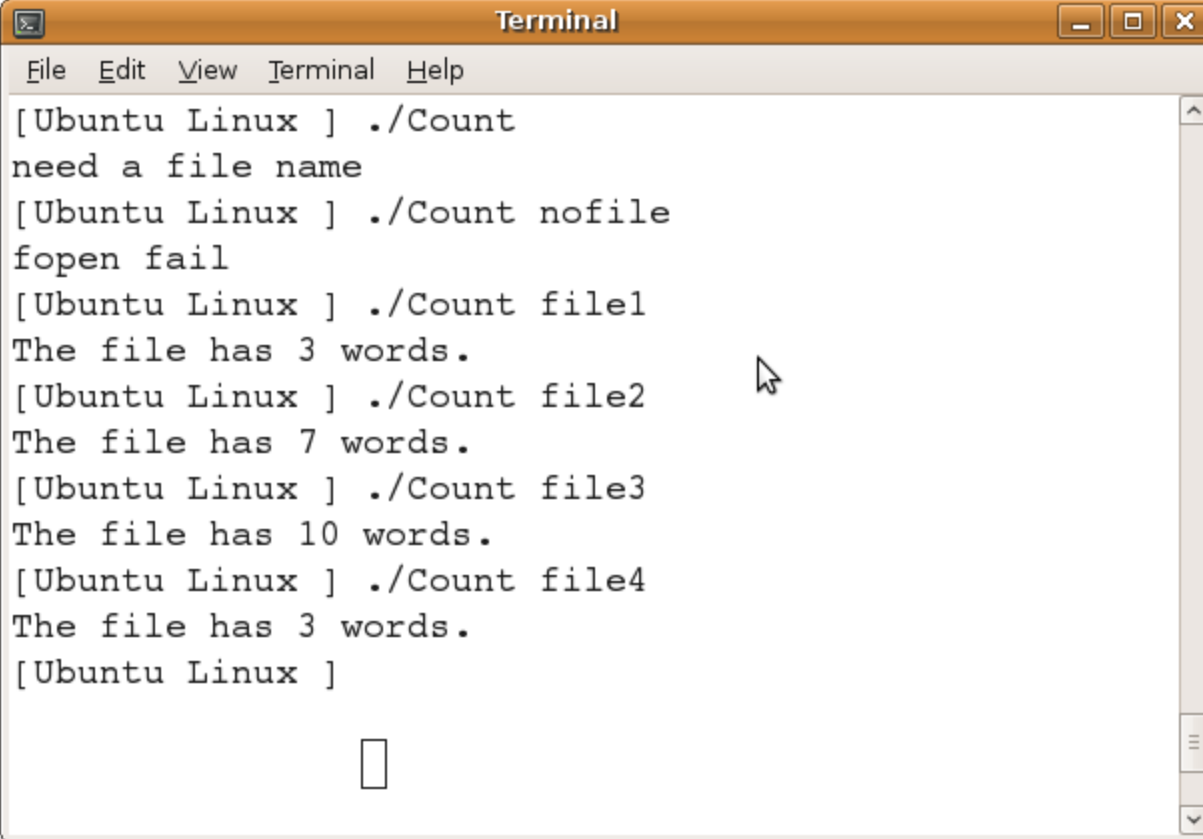
```
[Ubuntu Linux ] more file1
one two 3 4 5 three
[Ubuntu Linux ] more file2
one two 3 4 5 three
&^%a bcd^text file
[Ubuntu Linux ] more file3
one two 3 4 5 three
&^%a bcd^text file
C Programming          lesson
[Ubuntu Linux ] more file4
321 one two three
%
[Ubuntu Linux ]
```

3 words

3 + 4 = 7 words

7 + 3 = 10 words

Program's Output

A screenshot of a Linux terminal window titled "Terminal". The window has a menu bar with "File", "Edit", "View", "Terminal", and "Help". The terminal content shows the execution of a program named "Count". The user enters "./Count" and receives the message "need a file name". Then, the user enters "./Count nofile" and receives "fopen fail". Next, the user enters "./Count file1" and receives "The file has 3 words.". This is followed by "./Count file2" resulting in "The file has 7 words.", and then "./Count file3" resulting in "The file has 10 words.". Finally, the user enters "./Count file4" and receives "The file has 3 words.". The prompt "[Ubuntu Linux]" is visible at the end of the last line, with a cursor character below it. A mouse cursor is visible over the text "The file has 7 words.".

```
[Ubuntu Linux ] ./Count
need a file name
[Ubuntu Linux ] ./Count nofile
fopen fail
[Ubuntu Linux ] ./Count file1
The file has 3 words.
[Ubuntu Linux ] ./Count file2
The file has 7 words.
[Ubuntu Linux ] ./Count file3
The file has 10 words.
[Ubuntu Linux ] ./Count file4
The file has 3 words.
[Ubuntu Linux ]
```

```

#include <string.h>
int main(int argc, char * argv[])
{
    if (argc < 2)
    {
        printf("need a file name\n");
        return -1;
    }
    FILE * fh = fopen(argv[1], "r");
    if (fh == NULL)
    {
        printf("fopen fail\n");
        return -1;
    }
    int numWord = 0;
    int isWord = 0;
    int ch;
    while ((ch = fgetc(fh)) != EOF)
    {
        if (((ch >= 'a') && (ch <= 'z')) || ((ch >= 'A') && (ch <= 'Z')))
        {
            if (isWord == 0)
            {
                numWord ++;
            }
            isWord = 1;
        }
        else
        {
            isWord = 0;
        }
    }
    fclose(fh);
    printf("The file has %d words.\n", numWord);
    return 0;
}

```

read one character until
reaching the end of file

check whether
 $ch \in \{a, b, c, \dots, z\} \cup \{A, B, C, \dots, Z\}$

man fgetc(3) - Mozilla Firefox

File Edit View History Bookmarks Yahoo! Tools Help

<

>

↺

✕

🏠

📄

http://www.manpagez.com/man/3/fgetc/

☆

🔍

Google

DESCRIPTION

The **fgetc()** function obtains the next input character (if present) from the stream pointed at by *stream*, or the next character pushed back on the stream via **ungetc(3)**.

The **getc()** function acts essentially identically to **fgetc()**, but is a macro that expands in-line.

The **getchar()** function is equivalent to **getc(stdin)**.

The **getw()** function obtains the next *int* (if present) from the stream pointed at by *stream*.

The **getc_unlocked()** and **getchar_unlocked()** functions are equivalent to **getc()** and **getchar()** respectively, except that the caller is responsible for locking the stream with **flockfile(3)** before calling them. These functions may be used to avoid the overhead of locking the stream for each character, and to avoid input being dispersed among multiple threads reading from the same stream.

RETURN VALUES

If successful, these routines return the next requested object from the stream. Character values are returned as an *unsigned char* converted to an *int*. If the stream is at end-of-file or a read error occurs, the routines return EOF. The routines **feof(3)** and **ferror(3)** must be used to distinguish between end-of-file and error. If an error occurs, the global variable *errno* is set to indicate the error. The end-of-file con-

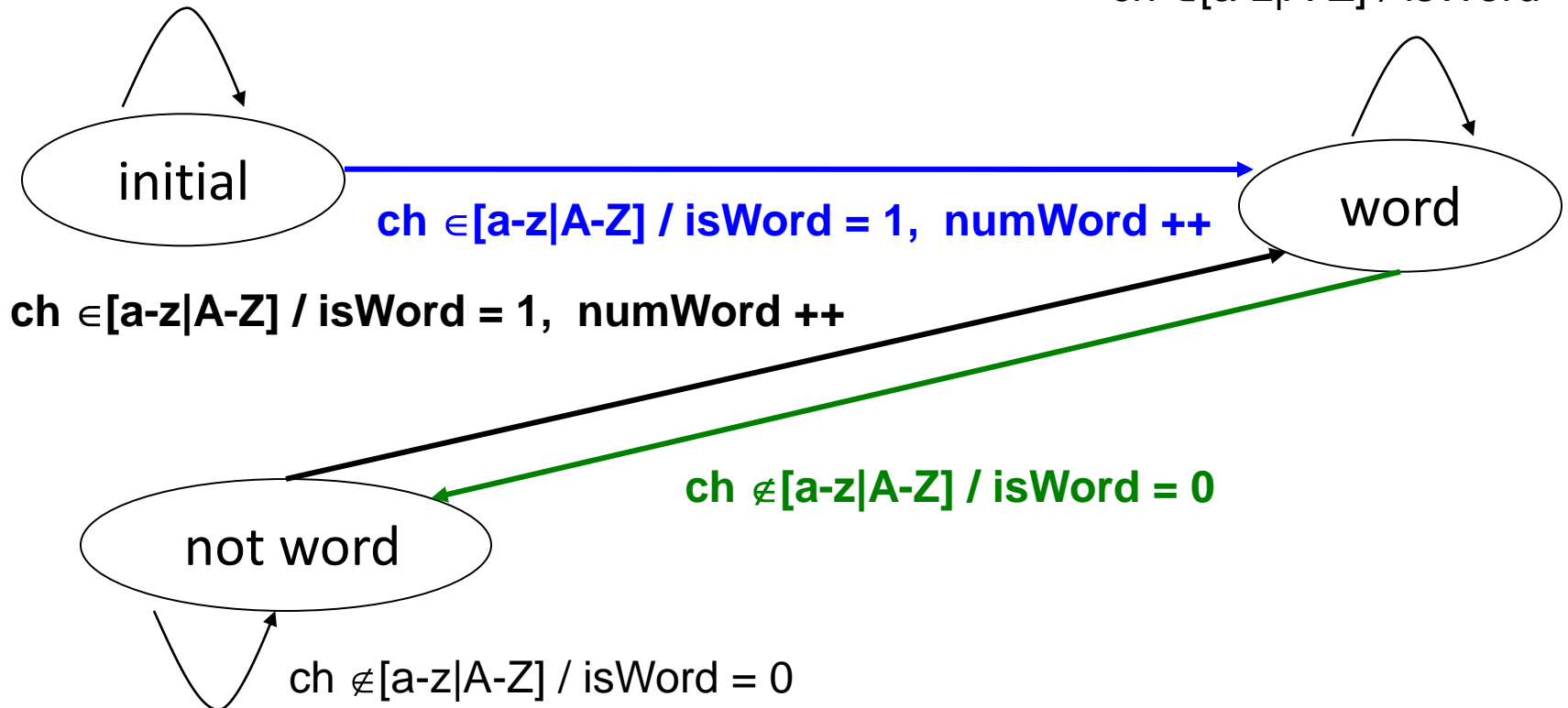
Done

State Diagram

condition / action
condition / action1, action2

$ch \notin [a-z|A-Z] / isWord = 0$

$ch \in [a-z|A-Z] / isWord = 1$



In the program, if
int isWord = 0;
is replaced by
int isWord = 1;
Which statement is true?

- ☒ A) The program will report 2 words in file1.
- ☐ B) The program will report 3 words in file1.
- ☐ C) The program will report 4 words in file1.
- ☐ D) The program will report 0 word in file1.

Correct - Click anywhere to
continue

Incorrect - Click anywhere to
continue

Your answer:

You did not answer this question

You must answer the question
before continuing

Submit

Clear

Count Word

Your Score	{score}
Max Score	{max-score}
Number of Quiz Attempts	{total-attempts}

Question Feedback/Review Information Will Appear Here

Continue

Review Quiz