```
$Id: lab4c-stdio-getopt.mm,v 1.6 2012-10-23 17:41:32-07 - - $
PWD: /afs/cats.ucsc.edu/courses/cmps012b-wm/Labs-cmps012m/lab4c-stdio-getopt
```

1. Overview

This lab will again use the command line and **`getopt`**(3). In addition, you will open, read, and close files using the facilities in **`stdio`**(3). Use the **`man`**(1) command to read each of the man pages as they are mentioned in this lab.

2. Program Specification

The program specification is given in the format of a Unix **`man`**(1) page.

NAME

bcat — concatenate and display files

SYNOPSIS

`bcat` [**`-mns`**] [*filename* ...]

DESCRIPTION

The **`bcat`** utility reads files in sequence and copies each file to **`stdout`**, with options that control display attributes.

OPTIONS

All options precede all operands and are scanned via **`getopt`**(3). The following options are supported :

`-m` In the style of **`more`**(1), a title is printed in front of each file. A title consists of exactly 5 lines : an empty line, a line of 64 colons, a line with the name of the file (a minus sign (**`-`**) is used for **`stdin),`** a line of 64 colons, an empty line.

`-n` Line numbers are displayed to the left of each line in a field of width 6 followed by 2 spaces.

`-s` Multiple empty lines are squeezed into a single empty line. That is, if three or more consecutive newline characters (**`\n`**) appear on input, only the first two are copied. They are still counted, though, for the purposes of printing line numbers if the **`-n`** option is specified.

OPERANDS

Each operand is the name of an input file. If no filenames are specified, **`bcat`** reads from **`stdin`** If a filename is given as a minus sign (**`-`**), **`stdin`** will be read at that point. The file **`stdin`** is never closed and multiple occurrences are accepted without complaint.

EXIT STATUS

0 Normal successful completion.

1 An error has occurred. Program execution continues if possible in the presence of an error.

SEE ALSO

`cat`(1), **`more`**(1), **`basename`**(3), **`errno`**(3), **`fclose`**(3), **`fopen`**(3), **`getc`**(3), **`getopt`**(3), **`putchar`**(3), **`strcmp`**(3), **`strerror`**(3).

3. Lab Sequence

Following is a suggested implementation sequence.

(1) Note the chapter number in parentheses after a reference to a command or function. The notation **`printf`**(3), for example, means the function **`printf`** in section 3 of the manual.

```
man -s 3 printf
```

(2) Make **`catfile`** print out a file title before going into a loop if the **`-m`** option is specified.

(3) Make it print line numbers in `"%6d␣␣"` format if the `-n` option is specified. Note that there are exactly two spaces after the format. The symbol ␣ is a visible space; that is, a graphic which represents a space character. To do this you will need a flag, which is initially true. Immediately after reading a character and determinint that it is not `EOF`, if the flag is set, print the line number and turn off the flag. Every time you see a newline, turn it back on.

(4) Your program should be compiled with the following, and no warnings should be produced:

```
gcc -g -O0 -Wall -Wextra -std=gnu99 bcat.c -o bcat
```

(5) Reading assignment: `/afs/cats.ucsc.edu/courses/cmps012b-wm/Coding-style/`

4. What to Submit

Submit `bcat.c` and `README` If you are doing pair programming also submit `PARTNER`