Programming Assignment 1

Due Tuesday, September 1 before midnight

NAME

`bcount` – count the occurrences of a specified byte in a file

USAGE

```
bcount path byte
```

DESCRIPTION

`bcount` counts the number of times the given 8-bit `byte` appears in the file `path`, and prints the result to `stdout`. It must use the UNIX `read()` system call to read the file.

`bcount` detects and reports (via `cerr` or `stderr`) the following errors:

- wrong command-line format (`bcount` prints a usage statement)
- invalid byte
- file cannot be opened

Use `perror()` to generate appropriate error messages, and `exit()` with a useful error return.

HINTS

See the `Public/os/bin` folder for a working version.

A byte is a number from 0 to 255, stored as an `unsigned char`. I suggest you use `atoi` to convert the command line string `byte` to a byte value, at first. However, `atoi` gives no error checking, so once things are working you should switch to the more robust `strtol`.

I'd suggest a blocksize of 4096 for your read buffer, but feel free to experiment. Remember to declare the appropriate sized array.

Look at the man pages for `open`, `read`, and `close` (especially the return values section). You'll want to open the file as read only. Check the return code – if it's -1, you can call `perror` to print the appropriate message. Otherwise, that return code is needed for your subsequent calls to `read` and `close`.

The `read` system call returns the number of bytes read. Keep calling read until 0 is returned (end-of-file), or -1 is returned (an error occurred). Don't scan all 4096 bytes in your buffer if read only filled 50 of them.

USEFUL MAN PAGES

`open(2)`	`errno(2)`
`close(2)`	`perror(3)`
`read(2)`	`strtol(3)`
`exit(2)`	`atoi(3)`