

Problem 1: System calls, error checking, and reporting

Caleb Zulawski

September 16, 2015

1 Implementation

```
$ ./copycat -h
Usage: copycat [OPTION]... [FILE]...
Concatenate FILE(s), or standard input, to standard output.
Similar to GNU cat.
```

```
-v          print diagnostic messages to standard error
-b SIZE     size of internal copy buffer, in bytes
-m MODE     file mode, in octal
-o FILE     output to FILE instead of standard output
-h          display this help and exit
```

This software was developed and tested on Linux 3.10.18 on an x86-64 netbook. The buffer size dependent performance is shown in Figure 1. The higher performance for larger buffers indicates that reading into and writing from the buffer is very fast compared to the process of setting up the `read()` or `write()`. The larger buffers seem to have similar performance because there is virtually no overhead compared to the amount being written.

2 Examples

2.1 Basic use

```
$ ./copycat
Hello World
Hello World
EOF
$
```

2.2 Concatenating from standard input and a file

```
$ echo And again... | ./copycat - copycat.c
And again...
```

```

/* copycat.c
 * Caleb Zulawski
 *
 * Entrance point of the program.
 */

#include "copycat.h"

int main(int argc, char* argv[]) {
    Options options;

    cc_parse_args(argc, argv, &options);
    cc_log(&options);
    cc_copy(&options);

    return 0;
}
$

```

2.3 Writing to a file, with error handling

```

$ touch outfile
$ chmod 000 outfile
$ echo nope | ./copycat -o outfile -
Error opening file outfile: Permission denied
$ sudo chmod 644 outfile
$ echo yup | ./copycat -o outfile -
$ cat outfile
yup
$

```

2.4 Bad input file behavior

```

$ echo Uh oh | ./copycat - nofile
Uh oh
Error opening file nofile: No such file or directory
$ echo Uh oh | ./copycat nofile -
Error opening file nofile: No such file or directory
$

```

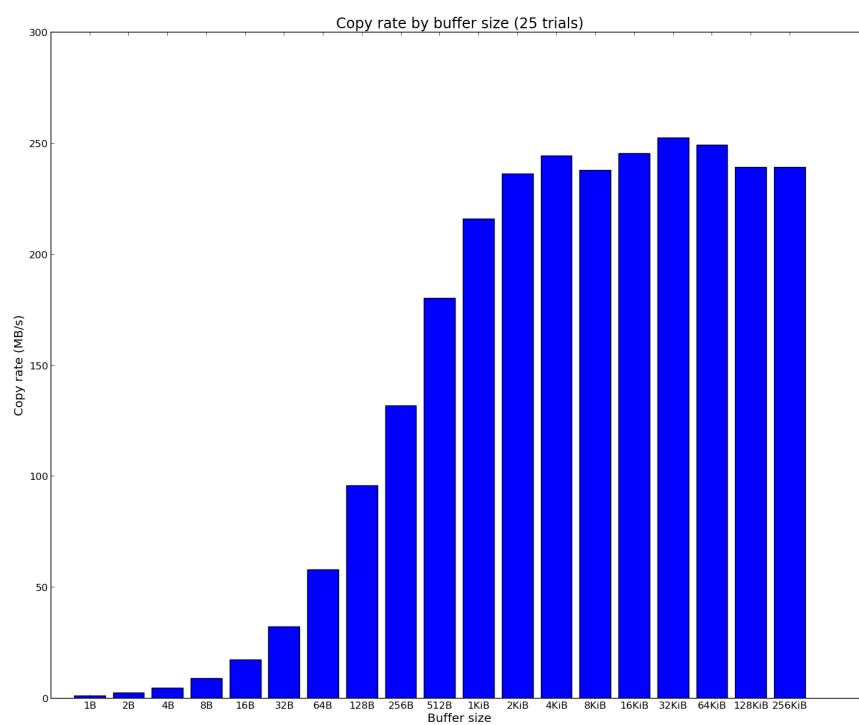


Figure 1: Performance