ECE-357 Assignment #1 Program Running Samples

Prof. Jeff Hakner
Di (David) Mei

Program Compiling:

```
gcc kitty.c –o kitty
```

Running Samples:

1. Concatenate standard input to standard output:

```
(base) Davids-MacBook-Pro:Desktop davidmei$ gcc kitty.c -o kitty
(base) Davids-MacBook-Pro:Desktop davidmei$ ./kitty
This is a standard input.
This is a standard input.
Hello World!
Hello World!
39 bytes transferred from file STANDARD_INPUT to file STANDARD_OUTPUT
Total number of read system calls: 3
Total number of write system calls: 2
(base) Davids-MacBook-Pro:Desktop davidmei$ ./kitty -
This is a standard input.
This is a standard input.
Hello World!
Hello World!
39 bytes transferred from file standard_input to file STANDARD_OUTPUT
Total number of read system calls: 3
Total number of write system calls: 2
(base) Davids-MacBook-Pro:Desktop davidmei$
```

2. Concatenate standard input to an output file:

```
(base) Davids-MacBook-Pro:Desktop davidmei$ ./kitty -o output.txt
This is a standard input.
Hello World!
39 bytes transferred from file STANDARD_INPUT to file output.txt
Total number of read system calls: 3
Total number of write system calls: 2
```

3. Open input files or concatenate input files to an output file:

```
(base) Davids-MacBook-Pro:Desktop davidmei$ ./kitty test1.txt test2.txt
abcdefg
hijklmn
16 bytes transferred from file test1.txt to file STANDARD_OUTPUT
opqrstu
vwxyz
14 bytes transferred from file test2.txt to file STANDARD_OUTPUT
Total number of read system calls: 4
Total number of write system calls: 2
```

```
(base) Davids-MacBook-Pro:Desktop davidmei$ ./kitty -o output.txt test1.txt test2.txt
16 bytes transferred from file test1.txt to file output.txt
14 bytes transferred from file test2.txt to file output.txt
Total number of read system calls: 4
Total number of write system calls: 2
```

4. Mixed concatenation & Binary file warning (binary_file is a sample binary file):

```
(base) Davids-MacBook-Pro:Desktop davidmei$ ./kitty -o output.txt - binary_file - test1.txt
This is a standard input.
26 bytes transferred from file STANDARD_INPUT to file output.txt
Warning: file binary_file is a binary file
1024 bytes transferred from file binary_file to file output.txt
This is another standard input.
32 bytes transferred from file STANDARD_INPUT to file output.txt
16 bytes transferred from file test1.txt to file output.txt
Total number of read system calls: 8
Total number of write system calls: 4
```

Failure Samples

1. Fail to open a file for reading:

```
(base) Davids-MacBook-Pro:Desktop davidmei$ gcc kitty.c -o kitty
(base) Davids-MacBook-Pro:Desktop davidmei$ ./kitty -o output.txt file_nonexist test1.txt
Error: unable to open file file_nonexist for reading (No such file or directory)
```

2. Fail to open a file for writing:

```
(base) Davids-MacBook-Pro:Desktop davidmei$ ./kitty -o / test1.txt test2.txt
Error: unable to open file / for writing (Is a directory)
```

3. Fail to write:

```
(base) Davids-MacBook-Pro:TOSHIBA$ ./kitty -o out.txt test3.txt
Error: unable to write file out.txt (No space left on device)
```

Reference:

Lines 29~40 in my code implements the method to overcome “partial writes”. I implemented the code by myself and the general idea about this method came from Stack Overflow:

https://stackoverflow.com/questions/32683086/handling-incomplete-write-calls