// **Program-1: open\_close.c**

#include <stdio.h>

#include <unistd.h>

#include <fcntl.h>

int main(int argc, char \*argv[]) {

int fd; /\* Open file \*/

if (argc == 1) {

printf("No file specified as command line argument.\n");

return (1);

}

if ((fd = open(argv[1], O\_RDWR)) == -1) {

printf("File opening.\n");

return(1);

}

else

printf("The file descriptor is %d.\n", fd);

/\* Close file \*/

if(close(fd) == -1) {

printf("File closing");

return(1);

}

else

printf("File closed successfully.\n");

return(0);

}

**//Output**

~/Week-5$ ./a.out foo.txt

File opening.

~/Week-5$ ./a.out foobar.txt

The file descriptor is 3.

File closed successfully.

|  |  |
| --- | --- |
|  |  |
|  | beza@beza:~/CS510-APUE/Homework/week04$ gcc hw1\_open\_close.c -o hw1  beza@beza:~/CS510-APUE/Homework/week04$ ./hw1 foo.txt  File opening.  beza@beza:~/CS510-APUE/Homework/week04$ touch foobar.txt  beza@beza:~/CS510-APUE/Homework/week04$ ./hw1 foobar.txt  The file descriptor is 3.  File closed successfully.  beza@beza:~/CS510-APUE/Homework/week04$ ls -a -l  total 28  drwxrwxr-x 2 beza beza 4096 May 23 10:54 .  drwxrwxr-x 6 beza beza 4096 May 23 10:52 ..  -rw-rw-r-- 1 beza beza 0 May 23 10:54 foobar.txt  -rwxrwxr-x 1 beza beza 16096 May 23 10:53 hw1  -rw-rw-r-- 1 beza beza 715 May 23 10:53 hw1\_open\_close.c  beza@beza:~/CS510-APUE/Homework/week04$ chmod 0364 foobar.txt  beza@beza:~/CS510-APUE/Homework/week04$ ./hw1 foobar.txt  File opening.  beza@beza:~/CS510-APUE/Homework/week04$ |

2) **Program-2:** **Open\_close\_rev.c**

#include <stdio.h>

#include <unistd.h>

#include <fcntl.h>

int main(int argc, char \*argv[]) {

int fd; /\* Open file \*/

if (argc == 1) {

printf("No file specified as command line argument.\n");

return (1);

}

close(0); /\* Closed standard input \*/

if ((fd = open(argv[1], O\_RDWR)) == -1) {

printf("File opening.\n");

return(1);

}

else

printf("The file descriptor is %d.\n", fd);

/\* Close file \*/

if(close(fd) == -1) {

printf("File closing");

return(1);

}

else

printf("File closed successfully.\n");

return(0);

}

**// Output**

~/Week-5$ ./a.out foo.txt

File opening.

~/Week-5$ ./a.out foobar.txt

The file descriptor is 0. // 要解释这里为什么是0

File closed successfully.

3) **Program-3: cpy.c**

// Copy the source file, an existing file, to the target file, a non-existent file.

// The compilation and execution of the example program shows the program output.

//

#include<fcntl.h>

#include <stdio.h>

#include <unistd.h>

#define SIZE 512

int closefd(int); // function prototype 要解释一下这个

int main(int argc, char \*argv[]) {

int rfd, wfd, nr, nw;

char buf[SIZE];

if(argc != 3) {

printf("Inappropriate number of command line arguments.\n");

return(0);

}

/\* Open the source file for reading \*/

if ((rfd = open(argv[1], O\_RDONLY)) == -1) {

printf("Source file opening");

return(1);

}

//Open the destination file for writing

if ((wfd = open(argv[2], O\_CREAT | O\_WRONLY | O\_TRUNC, 0666)) == -1) {

printf("Creation and opening of destination file\n");

return(1);}

while((nr = read(rfd, buf, sizeof(buf))) != 0) {

if (nr == -1) {

printf("File reading.\n");

return(1);

}

nw = write(wfd, buf, nr);

}

closefd(rfd);

closefd(wfd);

}

int closefd(int fd) {

if (close(fd) == -1) {

printf("File closing\n");

return(1);

}

}

**// Output:**

~/Week-5$ cc cpy.c

~/Week-5$ ./a.out cpy.c cpy\_bak.c

~/Week-5$ ls -l

total 180

-rwxr-xr-x 1 user user 16248 May 23 06:56 a.out

-rw-r--r-- 1 user user 4633 May 22 21:29 apue.h

-rw-r--r-- 1 user user 1039 May 22 22:35 cpy.c

-rw-r--r-- 1 user user 1039 May 23 06:57 cpy\_bak.c

-rw-r--r-- 1 user user 643 May 23 06:56 create\_holes.c

-rw-r--r-- 1 user user 931 May 23 06:30 create\_holes\_delete.c

-rw-r--r-- 1 user user 349807 May 22 22:51 data

-rw-r--r-- 1 user user 2069 May 23 05:30 data1

-rw-r--r-- 1 user user 16 May 22 21:36 foobar.txt

-rw-r--r-- 1 user user 503 May 22 21:37 open\_close.c

-rw-r--r-- 1 user user 544 May 22 22:01 open\_close\_rev.c

-rw-r--r-- 1 user user 835 May 23 05:28 truncate.c

|  |  |
| --- | --- |
|  |  |
|  | beza@beza:~/CS510-APUE/Homework/week04$ gcc hw2\_cpy.c -o hw2  beza@beza:~/CS510-APUE/Homework/week04$ ls  foobar.txt hw1 hw1\_open\_close.c hw2 hw2\_cpy.c  beza@beza:~/CS510-APUE/Homework/week04$ ./hw2 hw2\_cpy.c hw2\_cpy.bak  beza@beza:~/CS510-APUE/Homework/week04$ ls -a -l  total 52  drwxrwxr-x 2 beza beza 4096 May 23 11:22 .  drwxrwxr-x 6 beza beza 4096 May 23 10:52 ..  --wxrw-r-- 1 beza beza 0 May 23 10:54 foobar.txt  -rwxrwxr-x 1 beza beza 16096 May 23 10:53 hw1  -rw-rw-r-- 1 beza beza 715 May 23 10:53 hw1\_open\_close.c  -rwxrwxr-x 1 beza beza 16256 May 23 11:20 hw2  -rw-rw-r-- 1 beza beza 1369 May 23 11:22 hw2\_cpy.bak  -rw-rw-r-- 1 beza beza 1369 May 23 11:20 hw2\_cpy.c |

4) **Program-4: create\_holes.c P804**

#include <stdio.h>

#include <fcntl.h>

#include <unistd.h>

#include <string.h>

#define HSIZE 512L // Long 要解释

int main(int argc, char \*argv[])

{

int fd, i, n, nholes, nw;

char \*buf = "Random data for creating holes.";

/\* Open the given file \*/

if (argc != 3) {

printf("Inappropriate number of command line arguments \n");

return(0);

}

nholes = atoi(argv[2]); // 解释atoi

if ((fd = open(argv[1], O\_WRONLY)) == -1) {

printf("File opening\n");

return(1);

}

for (i = 0; i < nholes; i++) {

n = lseek(fd, HSIZE, SEEK\_END);

nw = write(fd, buf, strlen(buf));

}

if (close(fd) == -1) {

printf("File closing/\n");

return(1);

}

return(0);

}

**// Output:**

~/Week-5$ man bash > data

~/Week-5$ ls -l data

-rw-r--r-- 1 user user 348178 May 22 22:50 data

~/Week-5$ tail data | od

0000000 020012 020040 020040 020040 067503 070155 072557 062156

0000020 061440 066557 060555 062156 020163 067141 020144 067543

0000040 066555 067141 020144 062563 072561 067145 062543 020163

0000060 063157 072040 062550 063040 071157 020155 060540 035440

0000100 061040 035440 061440 020047 071141 020145 067556 020164

0000120 060550 062156 062554 020144 071147 061541 063145 066165

0000140 074554 073440 062550 020156 071160 061557 071545 020163

0000160 072563 070163 067145 064563 067157 064440 020163 072141

0000200 062564 070155 062564 027144 020040 064127 067145 020040

0000220 005141 020040 020040 020040 070040 067562 062543 071563

0000240 020040 071551 020040 072163 070157 062560 026144 072040

0000260 062550 071440 062550 066154 064440 066555 062145 060551

0000300 062564 074554 062440 062570 072543 062564 020163 064164

0000320 020145 062556 072170 061440 066557 060555 062156 064440

0000340 020156 064164 020145 062563 072561 067145 062543 020056

0000360 044440 020164 072563 063146 061551 071545 072040 020157

0000400 066160 061541 020145 064164 020145 062563 072561 067145

0000420 062543 067440 020146 067543 066555 067141 071544 061040

0000440 161145 110200 020012 020040 020040 020040 073564 062545

0000460 020156 060560 062562 072156 062550 062563 020163 067564

0000500 063040 071157 062543 064440 020164 067151 067564 060440

0000520 071440 061165 064163 066145 026154 073440 064550 064143

0000540 066440 074541 061040 020145 072163 070157 062560 020144

0000560 071541 060440 072440 064556 027164 005012 020040 020040

0000600 020040 040440 071162 074541 073040 071141 060551 066142

0000620 071545 066440 074541 067040 072157 024040 062571 024564

0000640 061040 020145 074145 067560 072162 062145 005056 020012

0000660 020040 020040 020040 064124 071145 020145 060555 020171

0000700 062542 067440 066156 020171 067157 020145 061541 064564

0000720 062566 061440 070157 067562 062543 071563 060440 020164

0000740 020141 064564 062555 005056 043412 052516 041040 071541

0000760 020150 027065 020061 020040 020040 020040 020040 020040

0001000 020040 020040 020040 020040 020040 020040 020040 020040

\*

0001040 020040 020040 020040 020040 020040 030062 030062 047440

0001060 072143 061157 071145 031040 020071 020040 020040 020040

0001100 020040 020040 020040 020040 020040 020040 020040 020040

\*

0001160 020040 040502 044123 030450 005051

0001172

~/Week-5$ cc -w create\_holes.c

~/Week-5$ ./a.out data 3

~/Week-5$ ls -l data

-rw-r--r-- 1 user user 349807 May 22 22:51 data

~/Week-5$ tail data | od

0000000 020040 020040 020040 041440 066557 067560 067165 020144

0000020 067543 066555 067141 071544 060440 062156 061440 066557

0000040 060555 062156 071440 070545 062565 061556 071545 067440

0000060 020146 064164 020145 067546 066562 060040 020141 020073

0000100 020142 020073 023543 060440 062562 067040 072157 064040

0000120 067141 066144 062145 063440 060562 062543 072546 066154

0000140 020171 064167 067145 070040 067562 062543 071563 071440

0000160 071565 062560 071556 067551 020156 071551 060440 072164

0000200 066545 072160 062145 020056 053440 062550 020156 060440

0000220 020012 020040 020040 020040 071160 061557 071545 020163

0000240 064440 020163 071440 067564 070160 062145 020054 064164

0000260 020145 064163 066145 020154 066551 062555 064544 072141

0000300 066145 020171 074145 061545 072165 071545 072040 062550

0000320 067040 074145 020164 067543 066555 067141 020144 067151

0000340 072040 062550 071440 070545 062565 061556 027145 020040

0000360 072111 071440 063165 064546 062543 020163 067564 070040

0000400 060554 062543 072040 062550 071440 070545 062565 061556

0000420 020145 063157 061440 066557 060555 062156 020163 062542

0000440 100342 005220 020040 020040 020040 072040 062567 067145

0000460 070040 071141 067145 064164 071545 071545 072040 020157

0000500 067546 061562 020145 072151 064440 072156 020157 020141

0000520 072563 071542 062550 066154 020054 064167 061551 020150

0000540 060555 020171 062542 071440 067564 070160 062145 060440

0000560 020163 020141 067165 072151 005056 020012 020040 020040

0000600 020040 071101 060562 020171 060566 064562 061141 062554

0000620 020163 060555 020171 067556 020164 074450 072145 020051

0000640 062542 062440 070170 071157 062564 027144 005012 020040

0000660 020040 020040 052040 062550 062562 066440 074541 061040

0000700 020145 067157 074554 067440 062556 060440 072143 073151

0000720 020145 067543 071160 061557 071545 020163 072141 060440

0000740 072040 066551 027145 005012 047107 020125 060502 064163

0000760 032440 030456 020040 020040 020040 020040 020040 020040

0001000 020040 020040 020040 020040 020040 020040 020040 020040

\*

0001040 020040 020040 020040 020040 031040 031060 020060 061517

0001060 067564 062542 020162 034462 020040 020040 020040 020040

0001100 020040 020040 020040 020040 020040 020040 020040 020040

\*

0001160 041040 051501 024110 024461 000012 000000 000000 000000

0001200 000000 000000 000000 000000 000000 000000 000000 000000

\*

0002160 000000 000000 000000 000000 051000 067141 067544 020155

0002200 060544 060564 063040 071157 061440 062562 072141 067151

0002220 020147 067550 062554 027163 000000 000000 000000 000000

0002240 000000 000000 000000 000000 000000 000000 000000 000000

\*

0003220 000000 000000 000000 000000 060522 062156 066557 062040

0003240 072141 020141 067546 020162 071143 060545 064564 063556

0003260 064040 066157 071545 000056 000000 000000 000000 000000

0003300 000000 000000 000000 000000 000000 000000 000000 000000

\*

0004260 000000 000000 000000 051000 067141 067544 020155 060544

0004300 060564 063040 071157 061440 062562 072141 067151 020147

0004320 067550 062554 027163

0004326

5) **//Program-5: create\_holes.c**

#include <stdio.h>

#include <fcntl.h>

#include <unistd.h>

#include <string.h>

#define HSIZE 512L

int main(int argc, char \*argv[])

{

int fd, i, n, nholes, nw;

char \*buf = "Random data for creating holes.";

/\* Open the given file \*/

if (argc != 3) {

printf("Inappropriate number of command line arguments \n");

return(0);

}

nholes = atoi(argv[2]);

if ((fd = open(argv[1], O\_WRONLY)) == -1) {

printf("File opening\n");

return(1);

}

for (i = 0; i < nholes; i++) {

n = lseek(fd, HSIZE, SEEK\_END);

nw = write(fd, buf, strlen(buf));

}

if (close(fd) == -1) {

printf("File closing/\n");

return(1);

}

return(0);

}

~/Week-5$ man bash > data1

~/Week-5$ ls -l data1

-rw-r--r-- 1 user user 2067 May 23 05:20 data1

~/Week-5$ tail data1

Copyright © 2020 Free Software Foundation, Inc. License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>.

This is free software: you are free to change and redistribute it. There is NO WARRANTY, to the extent permitted by law.

SEE ALSO

tac(1)

Full documentation <https://www.gnu.org/software/coreutils/cat>

or available locally via: info '(coreutils) cat invocation'

GNU coreutils 8.32 February 2024 CAT(1)

~/Week-5$ cc -w truncate.c

~/Week-5$ ./a.out data1 1024

The size of the original file is 2067 bytes.

The size of the truncated file is 1024 bytes.

~/Week-5$ tail data1

display TAB characters as ^I

-u (ignored)

-v, --show-nonprinting

use ^ and M- notation, except for LFD and TAB

--help display this help and exit

6) // **Program-6: truncate.c**

#include <fcntl.h>

#include<stdio.h>

#include <unistd.h>

#define HSIZE 512L

int main(int argc, char \*argv[]){

int fd, newfilesize, size\_o, size\_t; /\* Open the given file \*/

if (argc != 3) {

printf("Inappropriate number of command line arguments.\n");

return (0);

}

newfilesize = atoi(argv[2]);

if ((fd = open(argv[1], O\_WRONLY)) == -1) {

printf("File opening\n");

return (1);

}

size\_o = lseek(fd, 0, SEEK\_END); /\* Size of original file \*/

if (truncate(argv[1], newfilesize) == -1) {

printf("Truncating file\n");

return(1);

}

size\_t = lseek(fd,0,SEEK\_END); /\* Size of truncated file \*/

printf("The size of the original file is %d bytes.\n", size\_o);

printf("The size of the truncated file is %d bytes.\n", size\_t);

if (close(fd) == -1) {

printf("File closing \n");

return(1);

}

return(0);

}

**// Output**

~/Week-5$ ls

a.out apue.h cpy.c cpy\_bak.c create\_holes.c data data1 foobar.txt open\_close.c open\_close\_rev.c truncate.c

~/Week-5$ man cat > data1

~/Week-5$ ls -l data1

-rw-r--r-- 1 user user 2069 May 23 05:30 data1

~/Week-5$ tail data1

Copyright © 2020 Free Software Foundation, Inc. License GPLv3+: GNU GPL version 3 or later <https://gnu.org/licenses/gpl.html>.

This is free software: you are free to change and redistribute it. There is NO WARRANTY, to the extent permitted by law.

SEE ALSO

tac(1)

Full documentation <https://www.gnu.org/software/coreutils/cat>

or available locally via: info '(coreutils) cat invocation'

GNU coreutils 8.32

7) **create\_holes\_delete.c**

#include <fcntl.h>

#include <unistd.h>

#include <string.h>

#include <stdio.h>

#define HSIZE 512L

int main(int argc, char \*argv[]) {

int fd, i, n, nholes, nw, size\_o, size\_h;

char \*buf = "Random data for creating holes.";

/\* Open the given file \*/

if (argc != 3) {

printf("Inappropriate number of command line arguments.\n");

return (0);

}

nholes = atoi(argv[2]);

if ((fd = open(argv[1], O\_WRONLY)) == -1) {

printf("File opening\n");

return(1);

}

size\_o = lseek(fd,0,SEEK\_END); /\* Size of original file \*/

for (i = 0; i < nholes; i++) {

n = lseek(fd, HSIZE, SEEK\_END);

nw = write(fd, buf,strlen(buf));

}

size\_h = lseek(fd,0,SEEK\_END); /\* Size of file with holes \*/

if (close(fd) == -1) {

printf("File closing \n");

return(1);

}

printf("The size of the original file is %d bytes. \n", size\_o);

printf("The size of the file with holes is %d bytes.\n", size\_h);

unlink(argv[1]);

return(0);

}

**// Output**

~/Week-5$ man bash > data2

~/Week-5$ ls -l data2

-rw-r--r-- 1 user user 348178 May 23 06:33 data2

~/Week-5$ cc -w create\_holes\_delete.c

~/Week-5$ ./a.out data2 3

The size of the original file is 348178 bytes.

The size of the file with holes is 349807 bytes.