

/* File copy program. Error checking and reporting is minimal. */

```
#include <sys/types.h>                /* include necessary header files */
#include <fcntl.h>
#include <stdlib.h>
#include <unistd.h>
```

```
int main(int argc, char *argv[]);      /* ANSI prototype */
```

```
#define BUF_SIZE 4096                  /* use a buffer size of 4096 bytes */
#define OUTPUT_MODE 0700               /* protection bits for output file */
```

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

```
{
```

```
    int in_fd, out_fd, rd_count, wt_count;
    char buffer[BUF_SIZE];
```

```
    if (argc != 3) exit(1);            /* syntax error if argc is not 3 */
```

```
    /* Open the input file and create the output file */
```

```
    in_fd = open(argv[1], O_RDONLY);    /* open the source file */
```

```
    if (in_fd < 0) exit(2);              /* if it cannot be opened, exit */
```

```
    out_fd = creat(argv[2], OUTPUT_MODE); /* create the destination file */
```

```
    if (out_fd < 0) exit(3);            /* if it cannot be created, exit */
```

```
    /* Copy loop */
```

```
    while (TRUE) {
```

```
        rd_count = read(in_fd, buffer, BUF_SIZE); /* read a block of data */
```

```
        if (rd_count <= 0) break;          /* if end of file or error, exit loop */
```

```
        wt_count = write(out_fd, buffer, rd_count); /* write data */
```

```
        if (wt_count <= 0) exit(4);        /* wt_count <= 0 is an error */
```

```
    }
```

```
    /* Close the files */
```

```
    close(in_fd);
```

```
    close(out_fd);
```

```
    if (rd_count == 0)                  /* no error on last read */
```

```
        exit(0);
```

```
    else
```

```
        exit(5);                        /* error on last read */
```

```
}
```