Program 1: File Creation and Writing

#include <sys/types.h>

#include <sys/stat.h>

#include <fcntl.h>

#include <unistd.h>

#include <string.h>

#include <stdio.h>

int main() {

char filename[] = "example.txt";

int fileDescriptor;

// Open the file with write-only mode, creating if it doesn't exist

fileDescriptor = open(filename, O\_WRONLY | O\_CREAT, S\_IRUSR | S\_IWUSR);

if (fileDescriptor == -1) {

perror("Error opening the file");

return 1;

}

// Write a message to the file

char message[] = "Hello, World!";

ssize\_t bytesWritten = write(fileDescriptor, message, strlen(message));

if (bytesWritten == -1) {

perror("Error writing to the file");

close(fileDescriptor);

return 1;

}

printf("Message written to the file: %s\n", message);

// Close the file

close(fileDescriptor);

return 0;

}

Program 2: Writing to Standard Output (stdout)

#include <unistd.h>

#include <string.h>

#include <stdio.h>

int main() {

char message[] = "Hello, World!";

// Write the message to standard output (stdout)

ssize\_t bytesWritten = write(STDOUT\_FILENO, message, strlen(message));

if (bytesWritten == -1) {

perror("Error writing to stdout");

return 1;

}

return 0;

}

These programs demonstrate the use of the open() system call to create a file and the write() system call to write data to either a file or the standard output. The first program creates a file named "example.txt" and writes the message "Hello, World!" to it, while the second program writes the same message to the standard output.

Program 3: File Reading and Writing

#include <sys/types.h>

#include <sys/stat.h>

#include <fcntl.h>

#include <unistd.h>

#include <string.h>

#include <stdio.h>

int main() {

char sourceFile[] = "source.txt";

char destinationFile[] = "destination.txt";

int sourceDescriptor, destinationDescriptor;

char buffer[1024];

ssize\_t bytesRead, bytesWritten;

// Open the source file for reading

sourceDescriptor = open(sourceFile, O\_RDONLY);

if (sourceDescriptor == -1) {

perror("Error opening the source file");

return 1;

}

// Open or create the destination file for writing

destinationDescriptor = open(destinationFile, O\_WRONLY | O\_CREAT, S\_IRUSR | S\_IWUSR);

if (destinationDescriptor == -1) {

perror("Error opening/creating the destination file");

close(sourceDescriptor);

return 1;

}

// Read data from the source file and write it to the destination file

while ((bytesRead = read(sourceDescriptor, buffer, sizeof(buffer))) > 0) {

bytesWritten = write(destinationDescriptor, buffer, bytesRead);

if (bytesWritten == -1) {

perror("Error writing to the destination file");

close(sourceDescriptor);

close(destinationDescriptor);

return 1;

}

}

// Close the file descriptors

close(sourceDescriptor);

close(destinationDescriptor);

printf("File copied successfully from %s to %s.\n", sourceFile, destinationFile);

return 0;

}

Program 4: Standard Input Reading and File Writing

#include <sys/types.h>

#include <sys/stat.h>

#include <fcntl.h>

#include <unistd.h>

#include <string.h>

#include <stdio.h>

int main() {

char filename[] = "output.txt";

int fileDescriptor;

char buffer[1024];

ssize\_t bytesRead, bytesWritten;

// Open or create the output file for writing

fileDescriptor = open(filename, O\_WRONLY | O\_CREAT, S\_IRUSR | S\_IWUSR);

if (fileDescriptor == -1) {

perror("Error opening/creating the output file");

return 1;

}

// Read data from standard input and write it to the output file

while ((bytesRead = read(STDIN\_FILENO, buffer, sizeof(buffer))) > 0) {

bytesWritten = write(fileDescriptor, buffer, bytesRead);

if (bytesWritten == -1) {

perror("Error writing to the output file");

close(fileDescriptor);

return 1;

}

}

// Close the file descriptor

close(fileDescriptor);

printf("Data written to the file: %s\n", filename);

return 0;

}

Program 5: Appending Data to a File

#include <sys/types.h>

#include <sys/stat.h>

#include <fcntl.h>

#include <unistd.h>

#include <string.h>

#include <stdio.h>

int main() {

char filename[] = "data.txt";

int fileDescriptor;

char message[] = "Appending data...";

ssize\_t bytesWritten;

// Open the file in append mode

fileDescriptor = open(filename, O\_WRONLY | O\_APPEND);

if (fileDescriptor == -1) {

perror("Error opening the file");

return 1;

}

// Write the message to the file

bytesWritten = write(fileDescriptor, message, strlen(message));