// program of different system calls

#include <stdio.h>

#include <unistd.h>

#include <sys/wait.h>

#include <sys/types.h>

#include <fcntl.h>

#include <sys/stat.h>

#include <stdlib.h>

#include <string.h>

void process\_related() {

int choice;

pid\_t pid;

printf("\nProcess Related System Calls:\n");

printf("1. fork()\n");

printf("2. exit()\n");

printf("3. wait()\n");

printf("4. exec()\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch(choice) {

case 1:

pid = fork();

if (pid == 0) {

printf("This is the child process. PID = %d\n", getpid());

exit(0);

} else {

printf("This is the parent process. PID = %d\n", getpid());

}

break;

case 2:

printf("Exiting with status 0...\n");

exit(0);

break;

case 3:

pid = fork();

if (pid == 0) {

printf("Child process. PID = %d\n", getpid());

exit(0);

} else {

wait(NULL);

printf("Child process finished. Parent PID = %d\n", getpid());

}

break;

case 4:

pid = fork();

if (pid == 0) {

printf("Child process executing ls command:\n");

execl("/bin/ls", "ls", NULL);

exit(0);

} else {

wait(NULL);

printf("ls command executed by child process.\n");

}

break;

default:

printf("Invalid choice.\n");

break;

}

}

void communication\_related() {

int choice;

int fd[2];

char write\_msg[20] = "Hello, World!";

char read\_msg[20];

pid\_t pid;

printf("\nCommunication Related System Calls:\n");

printf("1. pipe()\n");

printf("2. FIFO (named pipe)\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch(choice) {

case 1:

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

perror("Pipe failed");

exit(1);

}

pid = fork();

if (pid == 0) {

close(fd[0]); // Close unused read end

write(fd[1], write\_msg, strlen(write\_msg)+1);

close(fd[1]);

} else {

close(fd[1]); // Close unused write end

read(fd[0], read\_msg, sizeof(read\_msg));

printf("Received message: %s\n", read\_msg);

close(fd[0]);

}

break;

case 2:

mkfifo("/tmp/myfifo", 0666);

pid = fork();

if (pid == 0) {

int fd = open("/tmp/myfifo", O\_WRONLY);

write(fd, write\_msg, strlen(write\_msg)+1);

close(fd);

exit(0);

} else {

int fd = open("/tmp/myfifo", O\_RDONLY);

read(fd, read\_msg, sizeof(read\_msg));

printf("Received message: %s\n", read\_msg);

close(fd);

}

unlink("/tmp/myfifo");

break;

default:

printf("Invalid choice.\n");

break;

}

}

void file\_related() {

int choice;

int fd;

char buffer[100];

printf("\nFile Related System Calls:\n");

printf("1. open(), write(), close()\n");

printf("2. link(), stat(), unlink()\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch(choice) {

case 1:

fd = open("example.txt", O\_WRONLY | O\_CREAT, 0644);

if (fd == -1) {

perror("Error opening file");

exit(1);

}

write(fd, "Hello, World!\n", 14);

close(fd);

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

fd = open("example.txt", O\_RDONLY);

read(fd, buffer, 100);

printf("File content: %s", buffer);

close(fd);

break;

case 2:

link("example.txt", "example\_link.txt");

struct stat file\_stat;

stat("example\_link.txt", &file\_stat);

printf("File size: %ld bytes\n", file\_stat.st\_size);

unlink("example\_link.txt");

printf("Link removed.\n");

break;

default:

printf("Invalid choice.\n");

break;

}

}

void info\_related() {

int choice;

printf("\nInformation Related System Calls:\n");

printf("1. alarm()\n");

printf("2. sleep()\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch(choice) {

case 1:

printf("Setting an alarm for 5 seconds...\n");

alarm(5);

sleep(6); // To show the alarm trigger

break;

case 2:

printf("Sleeping for 3 seconds...\n");

sleep(3);

printf("Woke up after 3 seconds.\n");

break;

default:

printf("Invalid choice.\n");

break;

}

}

int main() {

int choice;

while(1) {

printf("\nMenu:\n");

printf("1. Process Related\n");

printf("2. Communication Related\n");

printf("3. File Related\n");

printf("4. Information Related\n");

printf("5. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch(choice) {

case 1:

process\_related();

break;

case 2:

communication\_related();

break;

case 3:

file\_related();

break;

case 4:

info\_related();

break;

case 5:

printf("Exiting...\n");

exit(0);

default:

printf("Invalid choice.\n");

break;

}

}

return 0;

}