Experiment Number: 4

Problem Statement: Execution of advance Linux commands

NAME: Omkar Kamble ROLLNO: 2

CLASS: AI - B BATCH: 3

DATE OF PERFORMANCE: 7/08/23

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Code:

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <sys/types.h>

#include <sys/wait.h>

#include <sys/stat.h>

#include <fcntl.h>

#include <string.h>

**void** **processSystemCalls**() {

int status;

pid\_t child\_pid = **fork**();

**if** (child\_pid == 0) {

**printf**("Child process (PID: %d) executing...\n", **getpid**());

**exit**(0);

} **else** **if** (child\_pid > 0) {

**printf**("Parent process (PID: %d) waiting for child (PID: %d) to finish...\n", **getpid**(), child\_pid);

**wait**(&status);

**printf**("Child process (PID: %d) finished.\n", child\_pid);

} **else** {

**perror**("Fork failed");

}

}

**void** **fileSystemCalls**() {

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

**if** (fd != -1) {

char buffer[] = "Hello, this is a sample text.\n";

ssize\_t bytes\_written = **write**(fd, buffer, **strlen**(buffer));

**if** (bytes\_written > 0) {

**printf**("%ld bytes written to file.\n", bytes\_written);

} **else** {

**perror**("Write failed");

}

**close**(fd);

} **else** {

**perror**("Open failed");

}

//also read from the file

int fd2 = **open**("Sample.txt", O\_RDONLY);

**if** (fd2 != -1) {

char buffer[100];

ssize\_t bytes\_read = **read**(fd2, buffer, **sizeof**(buffer));

**if** (bytes\_read > 0) {

**printf**("%ld bytes read from file.\n", bytes\_read);

**printf**("File contents: %s\n", buffer);

} **else** {

**perror**("Read failed");

}

**close**(fd2);

} **else** {

**perror**("Open failed");

}

struct stat file\_info;

**if** (**stat**("Sample.txt", &file\_info) == 0) {

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

} **else** {

**perror**("Stat failed");

}

**if** (**unlink**("Sample.txt") == 0) {

**printf**("File 'sample\_file.txt' unlinked.\n");

} **else** {

**perror**("Unlink failed");

}

}

**void** **communicationSystemCalls**() {

int fd[2];

**if** (**pipe**(fd) == 0) {

char message[] = "Hello, this message is sent through a pipe.";

**write**(fd[1], message, **strlen**(message) + 1);

char received\_message[100];

**read**(fd[0], received\_message, **sizeof**(received\_message));

**printf**("Received message from pipe: %s\n", received\_message);

**close**(fd[0]);

**close**(fd[1]);

} **else** {

**perror**("Pipe failed");

}

}

**void** **informationSystemCalls**() {

**printf**("Current process ID: %d\n", **getpid**());

**printf**("Parent process ID: %d\n", **getppid**());

**printf**("User ID: %d\n", **getuid**());

**printf**("Effective User ID: %d\n", **geteuid**());

**printf**("Group ID: %d\n", **getgid**());

**printf**("Effective Group ID: %d\n", **getegid**());

}

int **main**() {

int choice;

**do** {

**printf**("\nMenu:\n");

**printf**("1. Process System Calls\n");

**printf**("2. File System Calls\n");

**printf**("3. Communication System Calls\n");

**printf**("4. Information System Calls\n");

**printf**("5. Exit\n");

**printf**("Enter your choice: ");

**scanf**("%d", &choice);

**switch** (choice) {

**case** 1:

**processSystemCalls**();

**break**;

**case** 2:

**fileSystemCalls**();

**break**;

**case** 3:

**communicationSystemCalls**();

**break**;

**case** 4:

**informationSystemCalls**();

**break**;

**case** 5:

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

**exit**(0);

default:

**printf**("Invalid choice. Please select a valid option.\n");

}

} **while** (1);

**return** 0;

}

**Output:**

**Menu**:

1. **Process** **System** **Calls**

2. **File** **System** **Calls**

3. **Communication** **System** **Calls**

4. **Information** **System** **Calls**

5. **Exit**

**Enter** your choice: 1

**Parent** process (PID: 1861) waiting **for** child (PID: 1904) to finish...

**Child** process (PID: 1904) executing...

**Child** process (PID: 1904) finished.

**Menu**:

1. **Process** **System** **Calls**

2. **File** **System** **Calls**

3. **Communication** **System** **Calls**

4. **Information** **System** **Calls**

5. **Exit**

**Enter** your choice: 2

30 bytes written to file.

30 bytes read **from** file.

**File** contents: **Hello**, this is a sample text.

**File** size: 30 bytes

**File** 'sample\_file.txt' unlinked.

**Menu**:

1. **Process** **System** **Calls**

2. **File** **System** **Calls**

3. **Communication** **System** **Calls**

4. **Information** **System** **Calls**

5. **Exit**

**Enter** your choice: 3

**Received** message **from** pipe: **Hello**, this message is sent through a pipe.

**Menu**:

1. **Process** **System** **Calls**

2. **File** **System** **Calls**

3. **Communication** **System** **Calls**

4. **Information** **System** **Calls**

5. **Exit**

**Enter** your choice: 4

**Current** process ID: 1861

**Parent** process ID: 1720

**User** ID: 1000

**Effective** **User** ID: 1000

**Group** ID: 1004

**Effective** **Group** ID: 1004

**Menu**:

1. **Process** **System** **Calls**

2. **File** **System** **Calls**

3. **Communication** **System** **Calls**

4. **Information** **System** **Calls**

5. **Exit**

**Enter** your choice: 5

**Exiting**...