Operating Systems (CS3000)

Lecture – 16

(Inter Process Communication - 5)

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY,
DESIGN AND MANUFACTURING,
KANCHEEPURAM

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY,
DESIGN AND MANUFACTURING,
KANCHEEPURAM

Dr. Jaishree Mayank

Assistant Professor

Department of Computer Sc. and Engg.

IPC using Named Pipe

- Communication between unrelated processes
- Execute client program from one terminal and the server program from another terminal.
- Named Pipe supports bi-directional communication.

To create a Named pipe

- int mknod(const char *pathname, mode_t mode, dev_t dev);
- create a special file or file system node such as ordinary file, device file, or FIFO.
- The pathname either absolute path or relative path of the file.
- The mode specified is the mode of file which specifies the file type and the file permission.
- The dev field is to specify device information such as major and minor device numbers (Default 0).
- return zero on success and -1 in case of failure.

Message Passing using Pipes

int mkfifo(const char *pathname, mode_t mode)

- a FIFO special file, which is used for named pipe.
- The file name can be either absolute path or relative path.
- The file mode information is as described as permission
- return zero on success and -1 in case of failure.

Message Passing using Named Pipes (FIFO)

- Algorithm: server process.

STEP1: Creates a named pipe (using library function mkfifo()) with name “fifofile” in directory, if not created. (Only one procees will create the pipe)

STEP2: Opens the named pipe for read purpose.

STEP3: Waits infinitely for a message from the client.

STEP4: Print message received from the client and close the file.

STEP5: Opens the named pipe for write purpose.

STEP6: Accepts string from the user.

STEP7: Sends a message to the client and close the named pipe.

STEP8: Repeats infinitely until the user enters the string “end”.

Message Passing using Named Pipes

- Algorithm: Client process

- **STEP1:** Opens the named pipe for write purpose.
- **STEP2:** Accepts string from the user.
- **STEP3:** Sends a message to the server and close the named pipe.
- **STEP4:** Open the named pipe for read purpose
- **STEP5:** Waits for the message from the server and prints the message.
- **STEP6:** Close the named pipe
- **STEP7:** Repeats infinitely until the user enters the string “end”.

```c
#include <stdio.h>
#include <string.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>
int main()
{
    int fd1;
    // FIFO file path
    char * myfifo = "/home/jaishree/Desktop/Desktop/OS/Process/myfile7.txt";
    // Creating the named file(FIFO)
    // mkfifo(<pathname>,<permission>)
    mkfifo(myfifo, 0666);
    char str1[80], str2[80];
    while (1)
    {
        // First open in read only and read
        //sleep(10);
        fd1 = open(myfifo,O_RDONLY);
        printf("read file fd %d\n", fd1);
        read(fd1, str1, 80);

        // Print the read string and close
        printf("Client Reading: %s\n", str1);
        close(fd1);

        // Now open in write mode and write
        // string taken from user.
        fd1 = open(myfifo,O_WRONLY);
        printf("write file fd %d\n", fd1);
        fgets(str2, 80, stdin);
        write(fd1, str2, strlen(str2)+1);
        close(fd1);
    }
    return 0;
}
```

```c
#include <stdio.h>
#include <string.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>
int main()
{
    int fd1;
    // FIFO file path
    char * myfifo = "/home/jaishree/Desktop/Desktop/OS/Process/myfile7.txt";
    // Creating the named file(FIFO)
    // mkfifo(<pathname>,<permission>)
    //mkfifo(myfifo, 0666);
    char str1[80], str2[80];
    while (1)
    {
        // First open in read only and read
        //sleep(10);
        fd1 = open(myfifo,O_RDONLY);
        printf("read file fd %d\n", fd1);
        read(fd1, str1, 80);

        // Print the read string and close
        printf("Client Reading: %s\n", str1);
        close(fd1);

        // Now open in write mode and write
        // string taken from user.
        fd1 = open(myfifo,O_WRONLY);
        printf("write file fd %d\n", fd1);
        fgets(str2, 80, stdin);
        write(fd1, str2, strlen(str2)+1);
        close(fd1);
    }
    return 0;
```

Thank You

Any Questions?