Dextutor

Program for IPC using pipe() function

Leave a Comment / Programs / By Baljit Singh Saini

The second method for IPC is using the pipe() function. Before writing a program for IPC using pipe() function let us first understand its working.

Syntax:

```
#include<unistd.h>
int pipe(int pipefd[2]);
```

pipe() function creates a unidirectional pipe for IPC. On success it return two file descriptors pipefd[0] and pipefd[1]. pipefd[0] is the reading end of the pipe. So, the process which will receive the data should use this file descriptor. pipefd[1] is the writing end of the pipe. So, the process that wants to send the data should use this file descriptor.

The program below creates a child process. The parent process will establish a pipe and will send the data to the child using writing end of the pipe and the child will receive that data and print on the screen using the reading end of the pipe.

//Q. Program to send a message from parent process to child process using pipe()

```
#include<stdio.h>
#include<unistd.h>
#include<sys/types.h>
#include<sys/wait.h>
int main()
{
int fd[2],n;
char buffer[100];
pid_t p;
pipe(fd); //creates a unidirectional pipe with two end fd[0] and fd[1]
p=fork();
if(p>0) //parent
{
printf("Parent Passing value to child\n");
write(fd[1],"hello\n",6); //fd[1] is the write end of the pipe
wait();
```

```
write(1,buffer,n);
}
}
```

How it works?

The parent process create a pipe using pipe(fd) call and then creates a child process using fork(). Then the parent sends the data by writing to the writing end of the pipe by using the fd[1] file descriptor. The child then reads this using the fd[0] file descriptor and stores it in buffer. Then the child prints the received data from the buffer onto the screen.

Output

```
baljit@baljit:~/cse325$ ./a.out
Passing value to child
Child printing the received value
hello
```

Figure1: pipe() output

Viva Questions on Program for IPC using pipe() function

Q1. Which kind of data channel is created by pipe() system call: Unidirectional or bidirectional?
Q2. What does the pipe() system call return on success?
Q3. What does the pipe() system call return on failure?
Q4. Why fork() is used in the above program?
Q5. Which process (parent or child) in the above code, is using the writing end of the pipe?

Video Link

Relevant Programs on Inter Process Communication

← Previous Post

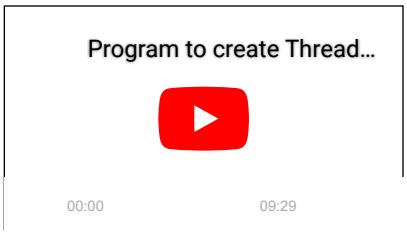
Next Post →

Leave a Comment

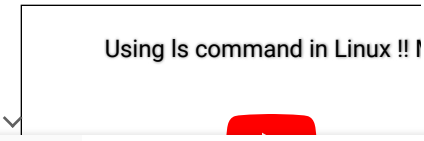
You must be logged in to post a comment.

Search ...

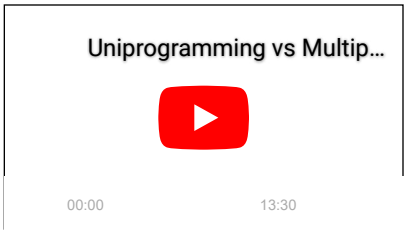
OS Lab



Linux Essentials Series



OS Theory



Categories

- Database
  - Entity-Relationship(ER)
  - Introduction to Database
- Linux
  - commands
  - Shell Scripting
- Operating System
  - CPU Scheduling
  - Deadlock
  - Disk Management
  - File Management
  - Introduction
  - IPC
  - Memory Management
  - Practice Problems
  - Process
  - Process Synchronization
  - System calls
  - Thread
- Programs
- Question Hub
- UGC-NET MCQ's for OS
- Uncategorized