dup() and dup2() Linux system call

Last Updated : 26 Sep, 2017

dup()

The dup() system call creates a copy of a file descriptor.

- It uses the lowest-numbered unused descriptor for the new descriptor.
- If the copy is successfully created, then the original and copy file descriptors may be used interchangeably.
- They both refer to the same open file description and thus share file offset and file status flags.

Syntax:

```
int dup(int oldfd);
oldfd: old file descriptor whose copy is to be created.
```

```
// CPP program to illustrate dup()
#include<stdio.h>
#include <unistd.h>
#include <fcntl.h>

int main()
{
    // open() returns a file descriptor file_desc to a
    // the file "dup.txt" here"

    int file_desc = open("dup.txt", O_WRONLY | O_APPEND);

    if(file_desc < 0)
        printf("Error opening the file\n");

    // dup() will create the copy of file_desc as the copy_desc
    // then both can be used interchangeably.
```

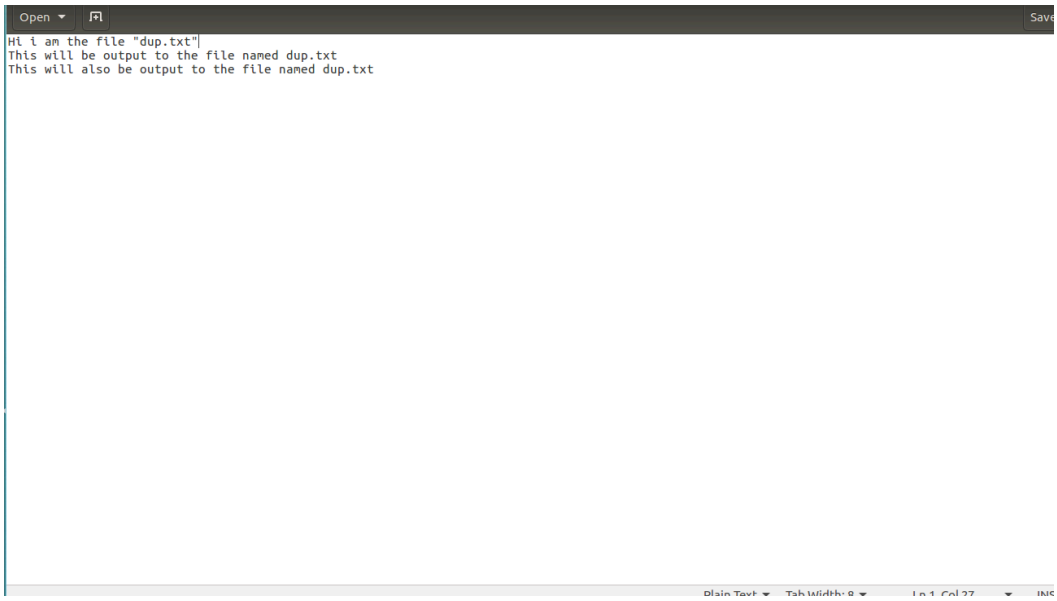
```
    int copy_desc = dup(file_desc);

    // write() will write the given string into the file
    // referred by the file descriptors

    write(copy_desc,"This will be output to the file named dup.txt\n",

    write(file_desc,"This will also be output to the file named dup.txt

    return 0;
}
```

Note that this program will not run in the online compiler as it includes opening a file and writing on it.

Explanation: The open() returns a file descriptor file_desc to the file named “dup.txt”. file_desc can be used to do some file operation with file “dup.txt”. After using the dup() system call, a copy of file_desc is created copy_desc. This copy can also be used to do some file operation with the same file “dup.txt”. After two write operations one with file_desc and another with copy_desc, same file is edited i.e. “dup.txt”.

Before running the code, Let The file “dup.txt” before the write operation be as shown below:

After running the C program shown above, the file “dup.txt” is as shown below:



dup2()

The dup2() system call is similar to dup() but the basic difference between them is that instead of using the lowest-numbered unused file descriptor, it uses the descriptor number specified by the user.

Syntax:

```
int dup2(int oldfd, int newfd);
```

oldfd: old file descriptor

newfd new file descriptor which is used by dup2() to create a copy.

Important points:

- Include the header file unistd.h for using dup() and dup2() system call.
- If the descriptor newfd was previously open, it is silently closed before being reused.
- If oldfd is not a valid file descriptor, then the call fails, and newfd is not closed.
- If oldfd is a valid file descriptor, and newfd has the same value as oldfd, then dup2() does
  nothing, and returns newfd.

A tricky use of dup2() system call: As in dup2(), in place of newfd any file

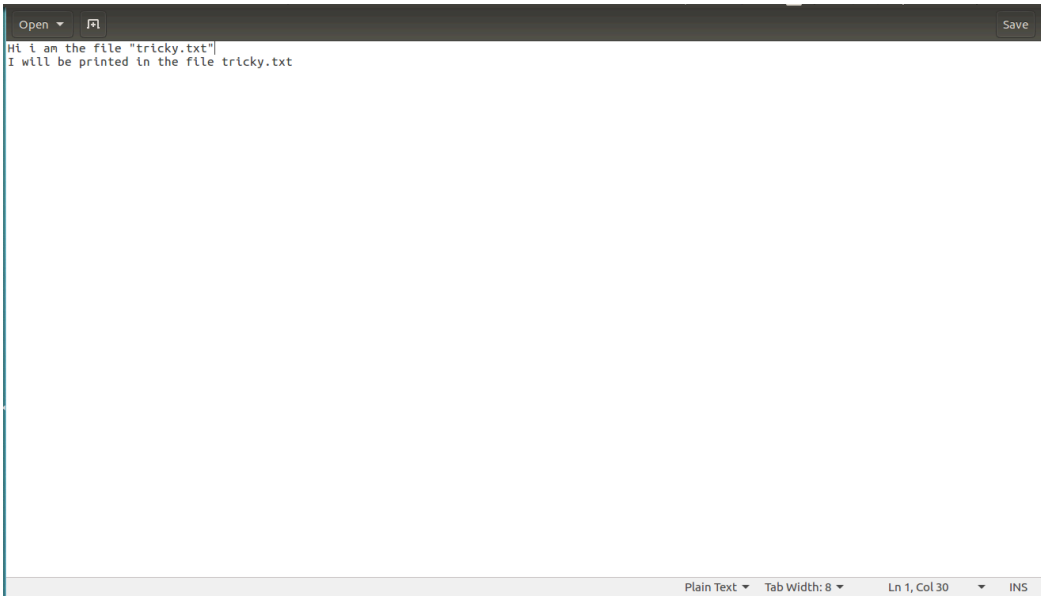
descriptor can be put. Below is a C implementation in which the file descriptor of Standard output (stdout) is used. This will lead all the printf() statements to be written in the file referred by the old file descriptor.

```
// CPP program to illustrate dup2()
#include<stdlib.h>
#include<unistd.h>
#include<stdio.h>
#include<fcntl.h>

int main()
{
    int file_desc = open("tricky.txt",O_WRONLY | O_APPEND);

    // here the newfd is the file descriptor of stdout (i.e. 1)
    dup2(file_desc, 1) ;

    // All the printf statements will be written in the file
    // "tricky.txt"
    printf("I will be printed in the file tricky.txt\n");

return 0;
}
```

This can be seen in the figure shown below:

Let The file “tricky.txt” before the dup2() operation be as shown below:

After running the C program shown above, the file “tricky.txt” is as shown below:



Reference: dup(2) – Linux man page

Get 90% Course fee refund on completing 90% course in 90 days!
Take the Three 90 Challenge today.

The next 90 Days of focus & determination can unlock your full potential. The Three 90 challenge has started and this is your chance to **upskill and get 90% refund.** What more motivation do you need? Start the challenge right away!

Comment | More info | Advertise with us

Next Article

Processes in Linux/Unix

Similar Reads

Accept system call

The accept() system call with the connection-based socket types(SOCK_STREAM, SOCK_SEQPACKET). It extracts the first connection...

4 min read

Linux Virtualization : Linux Containers (lxc)

Operating-system-level virtualization is a server virtualization method in which an operating system's kernel allows multiple isolated user-space...

6 min read

error: call of overloaded ‘function(x)’ is ambiguous | Ambiguity in...

Pre-requisite: Function Overloading in C++ Function overloading is a feature of object-oriented programming where two or more functions can have the...

9 min read

Difference between Management Information System (MIS) and Decisi...

Decision Support System (DSS) is an interactive, flexible computer based information system or sub-system intended to help decision makers use...

2 min read

Difference between Open-Loop Control System and Closed-Loop Contr...

Control System is a system in which the behavior of the system is determined by a differential equation. It manages the devices and the...

3 min read

Java System.nanoTime() vs System.currentTimeMillis

Java provides two methods to time operations, System.nanoTime() and System.currentTimeMillis(). But which one should be used in which...

3 min read

Useful and time saving bash commands in Linux

There are many useful bash scripts that can make things easier. Many things can be done just using the terminal. Here is a list of some of such useful ba...

3 min read

Compile and Run Java Programs in Sublime Text in Linux

Sublime Text is a free minimalist coding editor developed by Sublime HQ for desktop use. This development and IT program enable you to solely focus o...

2 min read

Network configuration and troubleshooting commands in Linux

Computers are often connected to each other on a network. They send requests to each other in the form of packets that travel from the host to th...

5 min read

Processes in Linux/Unix

A program/command when executed, a special instance is provided by the system to the process. This instance consists of all the services/resources...

6 min read

GeeksforGeeks

Corporate & Communications Address:

A-143, 7th Floor, Sovereign Corporate Tower, Sector- 136, Noida, Uttar Pradesh (201305)

Registered Address:

K 061, Tower K, Gulshan Vivante Apartment, Sector 137, Noida, Gautam Buddh Nagar, Uttar Pradesh, 201305

GET IT ON Google Play

Download on the App Store

Advertise with us

Company

About Us
Legal
Privacy Policy
Careers
In Media
Contact Us

Explore

Job-A-Thon Hiring Challenge
Hack-A-Thon
GfG Weekly Contest
Offline Classes (Delhi/NCR)
DSA in JAVA/C++
Master System Design

GFG Corporate Solution
Placement Training Program
Master CP
GeeksforGeeks Videos
Geeks Community

Languages
Python
Java
C++
PHP
GoLang
SQL
R Language
Android Tutorial

DSA
Data Structures
Algorithms
DSA for Beginners
Basic DSA Problems
DSA Roadmap
DSA Interview Questions
Competitive Programming

Data Science & ML
Data Science With Python
Data Science For Beginner
Machine Learning
ML Maths
Data Visualisation
Pandas
NumPy
NLP
Deep Learning

Web Technologies
HTML
CSS
JavaScript
TypeScript
ReactJS
NextJS
NodeJs
Bootstrap
Tailwind CSS

Python Tutorial
Python Programming Examples
Django Tutorial
Python Projects
Python Tkinter
Web Scraping
OpenCV Tutorial
Python Interview Question

Computer Science
GATE CS Notes
Operating Systems
Computer Network
Database Management System
Software Engineering
Digital Logic Design
Engineering Maths

DevOps
Git
AWS
Docker
Kubernetes
Azure
GCP
DevOps Roadmap

System Design
High Level Design
Low Level Design
UML Diagrams
Interview Guide
Design Patterns
OOAD
System Design Bootcamp
Interview Questions

School Subjects
Mathematics

Commerce
Accountancy

Physics
Chemistry
Biology
Social Science
English Grammar

Business Studies
Economics
Management
HR Management
Finance
Income Tax

Databases

SQL
MYSQL
PostgreSQL
PL/SQL
MongoDB

Preparation Corner

Company-Wise Recruitment Process
Resume Templates
Aptitude Preparation
Puzzles
Company-Wise Preparation
Companies
Colleges

Competitive Exams

JEE Advanced
UGC NET
UPSC
SSC CGL
SBI PO
SBI Clerk
IBPS PO
IBPS Clerk

More Tutorials

Software Development
Software Testing
Product Management
Project Management
Linux
Excel
All Cheat Sheets
Recent Articles

Free Online Tools

Typing Test
Image Editor
Code Formatters
Code Converters
Currency Converter
Random Number Generator
Random Password Generator

Write & Earn

Write an Article
Improve an Article
Pick Topics to Write
Share your Experiences
Internships

DSA/Placements

DSA - Self Paced Course
DSA in JavaScript - Self Paced Course
DSA in Python - Self Paced
C Programming Course Online - Learn C with Data Structures
Complete Interview Preparation
Master Competitive Programming
Core CS Subject for Interview Preparation
Mastering System Design: LLD to HLD
Tech Interview 101 - From DSA to System Design [LIVE]
DSA to Development [HYBRID]

Development/Testing

JavaScript Full Course
React JS Course
React Native Course
Django Web Development Course
Complete Bootstrap Course
Full Stack Development - [LIVE]
JAVA Backend Development - [LIVE]
Complete Software Testing Course [LIVE]
Android Mastery with Kotlin [LIVE]

Placement Preparation Crash Course [LIVE]

Machine Learning/Data Science

Complete Machine Learning & Data Science Program - [LIVE]
Data Analytics Training using Excel, SQL, Python & PowerBI - [LIVE]
Data Science Training Program - [LIVE]
Mastering Generative AI and ChatGPT
Data Science Course with IBM Certification

Programming Languages

C Programming with Data Structures
C++ Programming Course
Java Programming Course
Python Full Course

Clouds/Devops

DevOps Engineering
AWS Solutions Architect Certification
Salesforce Certified Administrator Course

GATE

GATE CS & IT Test Series - 2025
GATE DA Test Series 2025
GATE CS & IT Course - 2025
GATE DA Course 2025

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved