How to manage the priority of I / O process in Linux

If you've done this, you definitely notice that the system became less responsive during that time. On Linux, you can avoid this with the help of the ionice command.

Have you ever copied or moved dozens or hundreds of gigabytes of data? If you've done this, you definitely notice that the system became less responsive during that time. On Linux, you can avoid this with the help of the ionice command.

Manage priority of I / O process in Linux with ionice

1. What is the I / O priority level?
2. How does I / O priority work?
3. How to use the ionice command
    1. Ionice scheduling classes
    2. Useful ionice examples

What is the I / O priority level?

I / O stands for **Input / Output** (input / output). There are many types of I / O devices, but in this case, they are storage devices.

Each process that wants to read or write data to such a device is assigned a scheduling class and a priority number (or 'nice' value). This applies on Linux for file systems like ext4. Other file systems, such as ZFS, may perform slightly different methods to schedule read / write operations on the drive. In addition, the CFQ scheduler should be enabled for this to work. You can check with:

```
cat /sys/block/*/queue/scheduler
```

A process with a high 'nice' value has a lower priority.

How does I / O priority work?

An explicit storage device has a limited number of I / O operations that it can perform per second (IOPS). So when two processes want to read / write at the same time, each process will get an IOPS part. If they have the same priority, each process will receive about 50% IOPS.

But IOPS seems abstract and complex. For simplicity, you just need to think about the end result: Read / write speed. Suppose the drive can write to a maximum speed of 100MB / s. Procedure A begins a write operation. It writes to the drive at a speed of 100MB / s. Process B appears and wants to write to the same drive. It will write at about 50MB / s, and bring process A's write speed to the same value, 50MB / s. Now, if you give process B a higher nice I / O value, it will write at 20MB / s and let process A write at 80MB / s. When process A completes, process B will start writing at 100MB / s.

This example is very useful to understand something that can be confusing for some people. If a process has a very low priority (nice high value), it doesn't mean that the process will slow down every time. If it is the only process that uses the drive, it will read / write at maximum speed. But when other processes need the drive, it will temporarily exit and allow those processes to use more disk bandwidth. For a copy / write operation that can take hours to complete, assigning it to a low priority is a good idea, if you want to use your computer during that time.

How to use the ionice command

The general syntax of the command is:

```
ionice -c scheduling_class -n priority_nice_value command
```

Ionice scheduling classes

Idle (class 3) : The processes in this class are read / write only when no other program needs to access the drive. This means the read / write process is only at full speed when it is not in competition. When another program needs a disk, the process in the Idle class will only read / write with any remaining resources. From 100MB / s, it can temporarily write at 5MB / s, then return to 100MB / s when another program completes accessing the drive. This is the perfect class for long-term jobs that you don't want to slow down your system. There is no priority to specify for this class.

Example command:

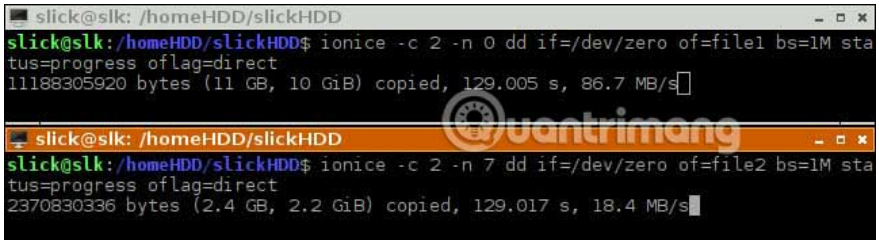
```
ionice -c 3 cp /home/user/largefile /Backups
```

Best-effort (class 2) : There is a nice priority / value between **0** and **7** . Remember, lower numbers mean higher priority. Use this class when you want to fine-tune disk access time for two or more processes.

For example, you want your backup to complete faster and give it a nice value of 0. Also, you're also transferring 6 movies to another drive but don't rush, so specify a nice value. for this process is 7.

Example command:

```
ionice -c 2 -n 0 backup_command
```



Realtime (class 1) : Only used if the process should be logged as soon as possible and not interrupted by any other very important program. Most users will never need this and should avoid using it, except in special cases. This class also supports nice values ??ranging from **0** to **7** . Only root can use this class. That means you will have to add the **sudo** prefix to the command.

Note that a process in the Realtime class of priority 0 can prevent other resource processes. In practical terms, that means other programs may have to wait several minutes or even hours to finish writing / reading a few megabytes of data. Use this class with care, only if you are sure you need it. If an important process in class 2 or 3 needs access to the drive, your system may freeze until the process in the Realtime class is finished.

Example command:

```
sudo ionice -c 1 -n 7 bash
```

Useful ionice examples

In the last example, instead of running the copy / move command, the shell was launched (Bash). Now, each subsequent command you enter into that shell will inherit the scheduling class I / O and priority. You can also do this on a graphical interface.

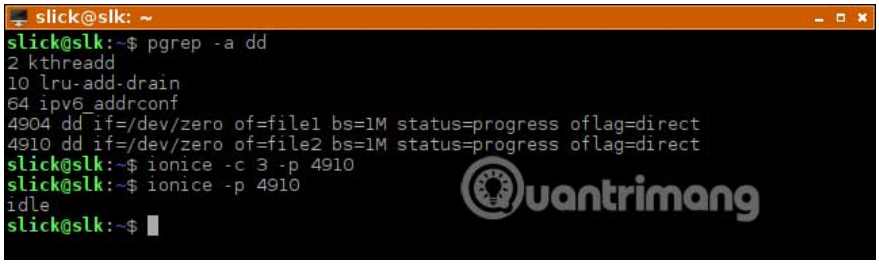
```
ionice -c 3 pcmanfm
```

The final command will launch a file explorer on the LXDE desktop environment. Replace '**pcmanfm**' with the name of the file explorer in your specific desktop. Now, all drive activity starting there will be done with the I / O Idle scheduling.

In other situations, the copy / move operation may have worked. In this case, you can use ionice in a different way.

```
ionice -c 3 -p 4910
```

This changes the priority class of the running program with process ID **4910.** You can find the PID (Process ID) with your task manager or with a command such as **pgrep.**



The ionice command can be useful on desktops that you don't want to experience lag while copying / moving large files. But keep in mind that ionice may be even more useful on servers. You certainly don't want a website you host as a visitor to experience lag, while making a full backup, right?

Hope you are succesful.