How to stop process using the sigstop signal in Linux

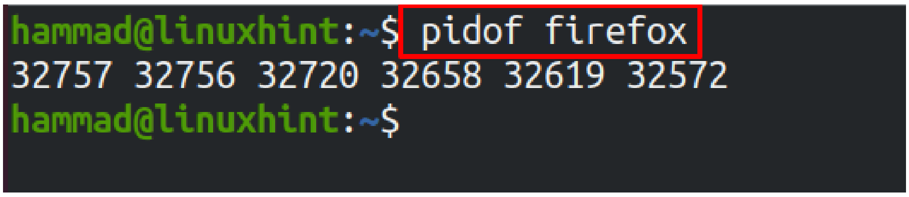
When a program or an application makes the overall processor halt, we try to terminate that program to get back the system to its normal condition. At this instant of time, we have two options, either to terminate that program using the "KILL" command or stop the process of the program using the "-SIGSTOP" signal with the KILL command.

The -SIGSTOP option is used with the KILL command, to stop the process, we will discuss it in detail in this write-up.

How to stop a process using the SIGSTOP signal in Linux

In Linux, the process is a running program or application which keeps the system busy for some time whereas an application/program is the executable files that are stored on the system. If an application is open, its process can be managed by using the KILL command; the KILL command has different signals like SIGSTOP, SIGCONT, SIGQUIT, and SIGTRAP to manage the processes. By default, if we use the KILL command, it will consider the SIGTERM signal and will terminate the running program and application. To understand it, we will open the Mozilla Firefox application in Linux, to have the PIDs as well as to confirm the running status of Firefox we will use the command:

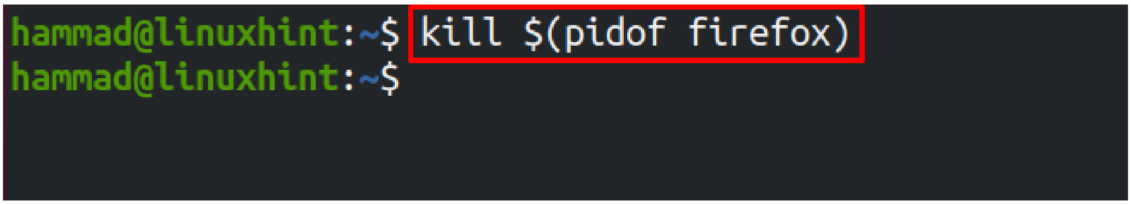
```
$ pidof firefox
```



```
hammad@linuxhint:~$ pidof firefox
32757 32756 32720 32658 32619 32572
hammad@linuxhint:~$
```

If Firefox is not running, it will show no output; the displaying of PIDs in the output has confirmed that Firefox is running. First, for the understanding, we will use the KILL command without any signal using Firefox:

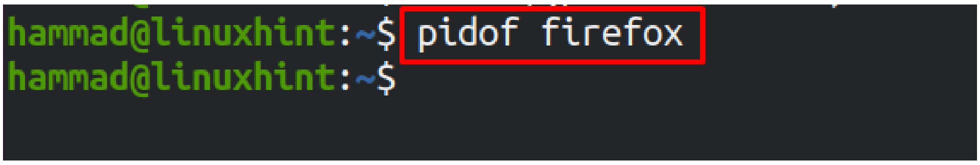
```
$ kill $(pidof firefox)
```



```
hammad@linuxhint:~$ kill $(pidof firefox)
hammad@linuxhint:~$
```

In the above command, Firefox has been terminated as we discussed before by default the kill command uses the SIGTERM signal, to confirm this we again use the PID command to check PIDs of Firefox:

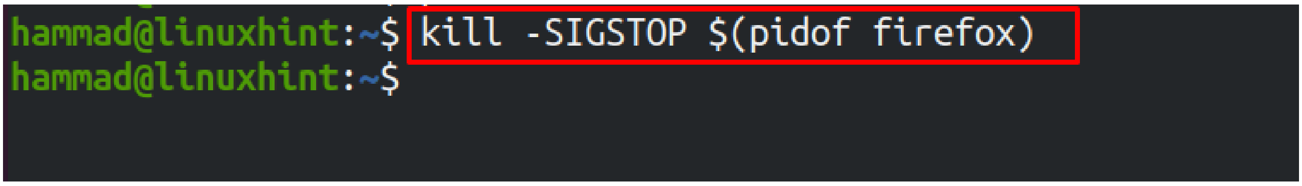
```
$ pidof firefox
```



```
hammad@linuxhint:~$ pidof firefox
hammad@linuxhint:~$
```

Nothing is displayed in the output that means the program has been terminated, to just stop the process, we will execute the command using the SIGSTOP signal after running Firefox again:

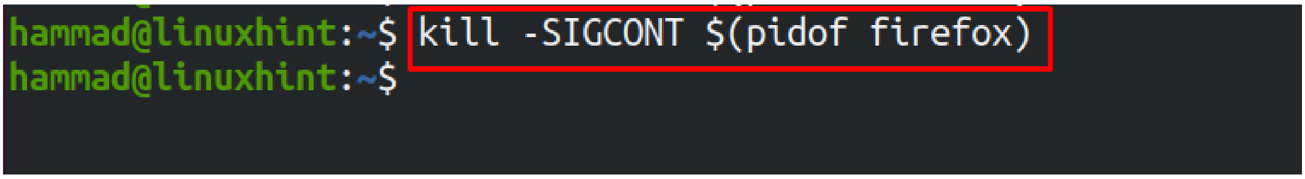
```
$ kill -SIGSTOP $(pidof firefox)
```



```
hammad@linuxhint:~$ kill -SIGSTOP $(pidof firefox)
hammad@linuxhint:~$
```

The output shows that the program has been put on hold for some time, instead of terminating it. To make the program normal again, execute the command:

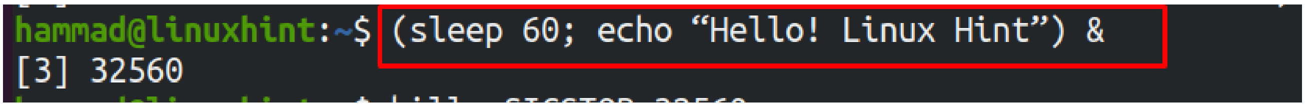
```
$ kill -SIGCONT $(pidof firefox)
```



```
hammad@linuxhint:~$ kill -SIGCONT $(pidof firefox)
hammad@linuxhint:~$
```

The SIGCONT signal continues the process from the time where it was made held stop, we can also use the PID to stop the process, for this we will write a script and display its PID using the “&” :

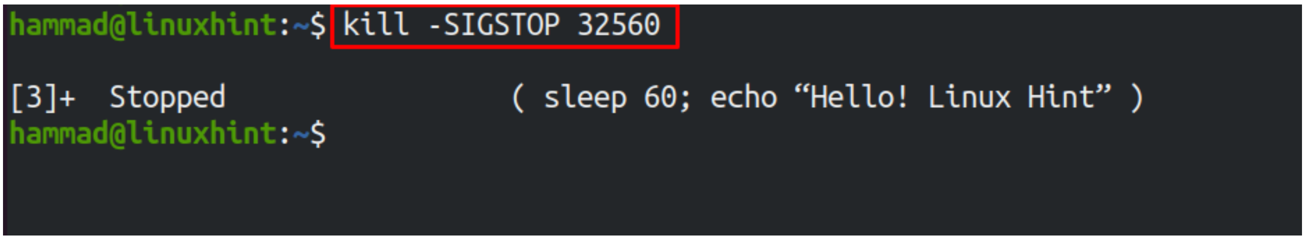
```
$ (sleep 60; echo “Hello! Linux Hint”) &
```



```
hammad@linuxhint:~$ (sleep 60; echo “Hello! Linux Hint”) &
[3] 32560
```

The above command will first sleep the terminal for 60 seconds and then will print “Hello! Linux Hint” but before it prints, we will hold the process using the “SIGSTOP” signal:

```
$ kill -SIGSTOP 32560
```

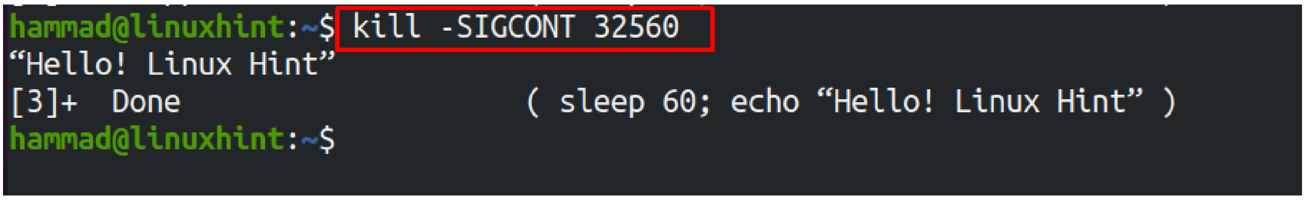


```
hammad@linuxhint:~$ kill -SIGSTOP 32560

[3]+  Stopped                 ( sleep 60; echo “Hello! Linux Hint” )
hammad@linuxhint:~$
```

The process of printing the script has been stopped, now to continue the process, use the "SIGCONT" signal:

```
$ kill -SIGCONT 32560
```



```
hammad@linuxhint:~$ kill -SIGCONT 32560
“Hello! Linux Hint”
[3]+  Done                    ( sleep 60; echo “Hello! Linux Hint” )
hammad@linuxhint:~$
```

The output has been displayed.

Conclusion

The sigstop is a signal which is used with the KILL command to stop the process for some time instead of terminating it permanently; the process can be resumed by using the sigcont command. This signal is very helpful, especially when some processes are making the system slow, we can stop these processes using the sigstop command. In this write-up, we have discussed the sigstop signal usage in detail with the help of examples. We stopped the processes with a sigstop signal using the program name as well as its PID for better understanding.