

**“AZƏRBAYCAN HAVA YOLLARI” CJSC NATIONAL AVIATION ACADEMY**

**Individual Work № : 9**

**Topic:** **C program to get the system information**

**Subject: OS2**

**Teacher: Mamed Shahmaliyev**

**Group: 1459i Student: Mehrali Babayev**

**Date: Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_**

**­ Baku 2022**

Getting System and Process Information Using C Programming and Shell in Linux

Whenever you start a new process in Linux it creates a file in /proc/ folder with the same name as that of the process id of the process. In that folder, there is a file named “status” which has all the details of the process. We can get those Process Information Through shell as follows:

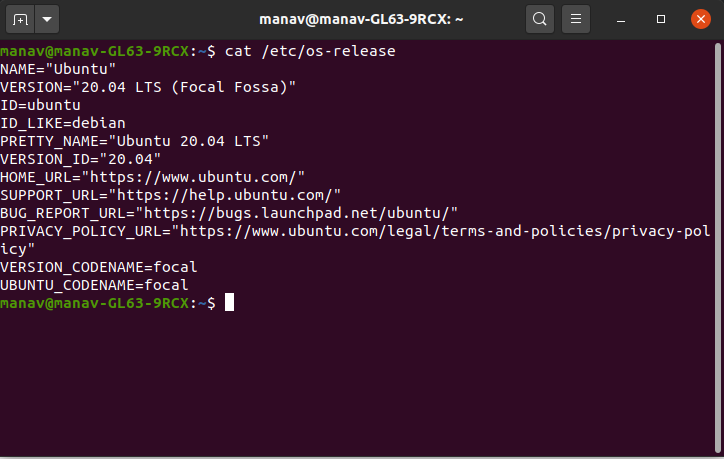
***cat /proc/1/status***

As can be seen, it displays most of the information about the process.

*Note:* In this case, the process id is 1, it may be changed as per need.

You can get the System Information through the shell. The basic system information is stored in a file named os-release in /etc/ folder.

***cat /etc/os-release***



You can also get the System Information using C programming. The below code is used to get the details of the system. In this code, utsname maintains a structure that has the details of the system like sysname nodename, release, version, etc.

***#include<stdio.h>***

***#include<stdlib.h>***

***#include<errno.h>***

***#include<sys/utsname.h>***

***int main()***

***{***

***struct utsname buf1;***

***errno =0;***

***if(uname(&buf1)!=0)***

***{***

***perror("uname doesn't return 0, so there is an error");***

***exit(EXIT\_FAILURE);***

***}***

***printf("System Name = %s\n", buf1.sysname);***

***printf("Node Name = %s\n", buf1.nodename);***

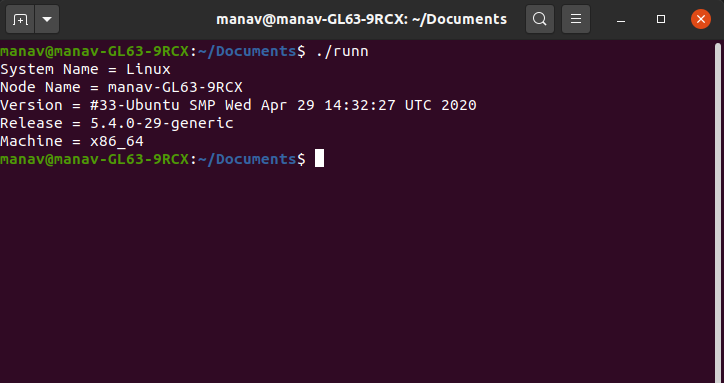
***printf("Version = %s\n", buf1.version);***

***printf("Release = %s\n", buf1.release);***

***printf("Machine = %s\n", buf1.machine);***

***}***

On execution the above code will give the following output:



To get Process Information using C programming, use the below code. In this code, we execute the Linux command through a c program to get the details of the process.

***#include<stdio.h>***

***#include<stdlib.h>***

***int main()***

***{ int r=system("cat /proc/1/status"); }***

On execution the above code will give the following output:

