Assignment 2

Operating System

Usama Sarwar

2019

Table of Contents

[I. Process 2](#_Toc23156703)

[II. Running Processes 2](#_Toc23156704)

[III. Processors in my Computer 3](#_Toc23156705)

[IV. Code for Process... 3](#_Toc23156706)

[Output 5](#_Toc23156707)

List of Fugures

[1. Task Manager of PC 2](#_Toc23156812)

[2. Processors of PC 3](#_Toc23156813)

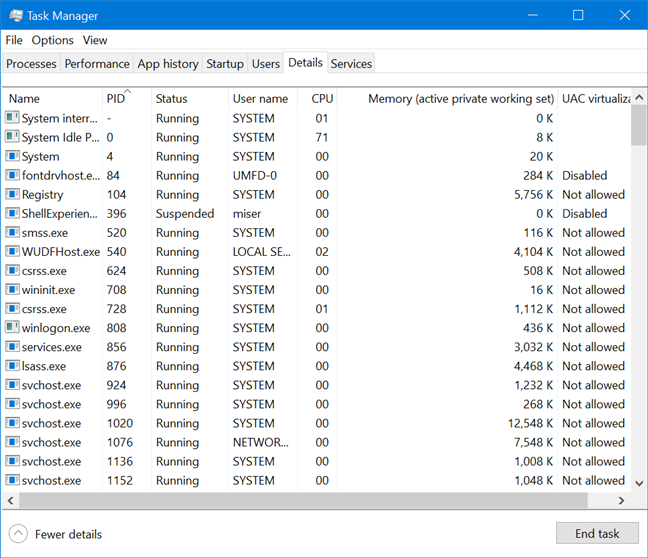
[3. Screenshoot of Console 5](#_Toc23156814)

# Process

In computing, a process is the instance of a computer program that is being executed by one or many threads. It contains the program code and its activity. Depending on the operating system, a process may be made up of multiple threads of execution that execute instructions concurrently.

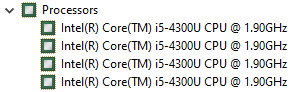
# Running Processes

Although Task Manager's Processes tab offers detailed information about how programs use system resources, it is the Details tab that allows you to find out everything you need to know about running processes (and more). Task Manager's Details tab provides generous data about each process running on your Windows 10 computer or device, and it can come in handy during advanced troubleshooting. In this tutorial, we go over the massive amount of information that it offers and what it can do



1. Task Manager of PC

# Processors in my Computer



2. Processors of PC

# Code for Process...

// Java program to control the Main Thread

public class Test extends Thread

{

public static void main(String[] args)

{

// getting reference to Main thread

Thread t = Thread.currentThread();

// getting name of Main thread

System.out.println("Current thread: " + t.getName());

// changing the name of Main thread

t.setName("Geeks");

System.out.println("After name change: " + t.getName());

// getting priority of Main thread

System.out.println("Main thread priority: "+ t.getPriority());

// setting priority of Main thread to MAX(10)

t.setPriority(MAX\_PRIORITY);

System.out.println("Main thread new priority: "+ t.getPriority());

for (int i = 0; i < 5; i++)

{

System.out.println("Main thread");

}

// Main thread creating a child thread

ChildThread ct = new ChildThread();

// getting priority of child thread

// which will be inherited from Main thread

// as it is created by Main thread

System.out.println("Child thread priority: "+ ct.getPriority());

// setting priority of Main thread to MIN(1)

ct.setPriority(MIN\_PRIORITY);

System.out.println("Child thread new priority: "+ ct.getPriority());

// starting child thread

ct.start();

}

}

// Child Thread class

class ChildThread extends Thread

{

@Override

public void run()

{

for (int i = 0; i < 5; i++)

{

System.out.println("Child thread");

}

}

}

## Output



3. Screenshoot of Console