Practical: - 5

Aim: -To write a c program to simulate the CPU scheduling round-robin algorithm.

Theory: -

Round-robin is a CPU scheduling algorithm that shares equal portions of resources in circular orders to each process and handles all processes without prioritization. In the round-robin, each process gets a fixed time interval of the slice to utilize the resources or execute its task called time quantum or time slice. Some of the round-robin processes are pre-empted if it executed in a given time slot, while the rest of the processes go back to the ready queue and wait to run in a circular order with the scheduled time slot until they complete their task. It removes the starvation for each process to achieve CPU scheduling by proper partitioning of the CPU.

Algorithm: -

1. The queue structure in ready queue is of First In First Out (FIFO) type.

2. A fixed time is allotted to every process that arrives in the queue. This fixed time is known as time slice or time quantum.

3. The first process that arrives is selected and sent to the processor for execution. If it is not able to complete its execution within the time quantum provided, then an interrupt is generated using an automated timer.

4. The process is then stopped and is sent back at the end of the queue. However, the state is saved and context is thereby stored in memory. This helps the process to resume from the point where it was interrupted.

5. The scheduler selects another process from the ready queue and dispatches it to the processor for its execution. It is executed until the time Quantum does not exceed.

6. The same steps are repeated until all the process are finished.

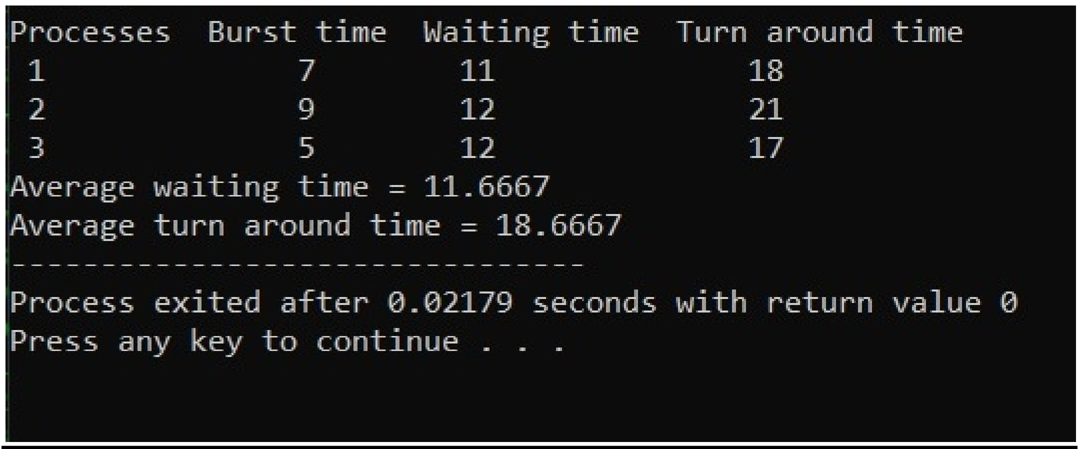
The round robin algorithm is simple and the overhead in decision making is very low. It is the best scheduling algorithm for achieving better and evenly distributed response time.

Code: -

```
#include<iostream>
using namespace std;
void findWaitingTime(int processes[], int n,
            int bt[], int wt[], int quantum)
{
    int rem_bt[n];

    for (int i = 0 ;i< n ; i++)
rem_bt[i] = bt[i];
    int t = 0;
    while (1)
    {
        bool done = true;
        for (int i = 0 ;i< n; i++)
        {
            if (rem_bt[i] > 0)
            {
                done = false;

                if (rem_bt[i] > quantum)
                {
                    t += quantum;
rem_bt[i] -= quantum;
                }
                else
                {
                    t = t + rem_bt[i];
wt[i] = t - bt[i];
rem_bt[i] = 0;
                }
            }
        }
        if (done == true)
        break;
    }
}
void findTurnAroundTime(int processes[], int n,
int bt[], int wt[], int tat[])
{
    for (int i = 0; i<n ;i++)
tat[i] = bt[i] + wt[i];
}
void findavgTime(int processes[], int n, int bt[],
                                    int quantum)
{
    int wt[n], tat[n], total_wt = 0, total_tat = 0;
findWaitingTime(processes, n, bt, wt, quantum);
findTurnAroundTime(processes, n, bt, wt, tat);
cout<< "Processes "<< " Burst time "
<< " Waiting time " << " Turn around time\n";

    for (int i=0; i<n; i++)
    {
total_wt = total_wt + wt[i];
total_tat = total_tat + tat[i];
```

```
cout<< " " << i+1 << "\t\t" <<bt[i] <<"\t "
<<wt[i] <<"\t\t " << tat[i] <<endl;
   }

cout<< "Average waiting time = "
<< (float)total_wt / (float)n;
cout<< "\nAverageturn around time = "
<< (float)total_tat / (float)n;
}
int main()
{
   int processes[] = { 1, 2, 3};
   int n = sizeof processes / sizeofprocesses[0];
  int burst_time[] = {10, 5, 8};
int quantum = 2;
findavgTime(processes, n, burst_time, quantum);
   return 0;
}
```

Output: -



```
Processes  Burst time  Waiting time  Turn around time
 1              7        11              18
 2              9        12              21
 3              5        12              17
Average waiting time = 11.6667
Average turn around time = 18.6667
--------------------------------
Process exited after 0.02179 seconds with return value 0
Press any key to continue . . .
```

Practical :- 6

Aim:- Introduction about virtualization , virtual box and installation of virtual machine.

OBJECTIVE:-

- What is virtualization
- What is virtual box
- Installation of virtual machine

- **What is virtualization:-** Virtualization refers to the act of creating a virtual (rather than actual) version of something, including (but not limited to) a virtual computer hardware platform, operating system (OS), storage device, or computer network resources. Virtualization began in the 1960s, as a method of logically dividing the system resources provided by mainframe computers between different applications. Since then, the meaning of the term has broadened.

TYPES:-

(1)Hardware virtualization or platform virtualization refers to the creation of a virtual machine that acts like a real computer with an operating system. Software executed on these virtual machines is separated from the underlying hardware resources. For example, a computer that is running Microsoft Windows may host a virtual machine that looks like a computer with the Ubuntu Linux operating system; Ubuntu-based software can be run on the virtual machine.

(2)Desktop virtualization is the concept of separating the logical desktop from the physical machine. One form of desktop virtualization, virtual desktop infrastructure (VDI), can be thought of as a more advanced form of hardware virtualization. Rather than interacting with a host computer directly via a keyboard, mouse, and monitor, the user interacts with the host computer using another desktop computer or a mobile device by means of a network connection, such as a LAN, Wireless LAN or even the Internet.

(3) Nested virtualization refers to the ability of running a virtual machine within another, having this general concept extendable to an arbitrary depth. In other words, nested virtualization refers to running one or more hypervisors inside another hypervisor. Nature of a nested guest virtual machine does not need not be homogenous with its host virtual machine; for example, application virtualization can be deployed within a virtual machine created by using hardware virtualization.

- **What is virtual box**

VirtualBox

VirtualBox Logo since 2010

RunningKubuntuLive CDwithOracleVM VirtualBox onWindows 7

Original author(s)	Innotek GmbH
Developer(s)	Oracle Corporation
Initial release	15 January 2007; 8 years ago

Stable release	4.3.26[1](16 March 2015; 14 days ago)[±]
Written in	C,C++
Operating system	Microsoft Windows,Mac OS X,LinuxandSolaris[2]
Size	86–115MBdepending on platform[3]
Type	Virtual machine
License	Base Package (USBsupport only forUSB 1.1):GNUGeneral Public Licenseversion 2 (OptionallyCDDLfor most files of the source distribution), "Extension Pack" (includingUSB 2.0support):PUEL
Website	www.virtualbox.org

Fig 2.1

• Installation of virtual machine

STEP 1:-Download virtual box oracle is available for free from developers website**.**

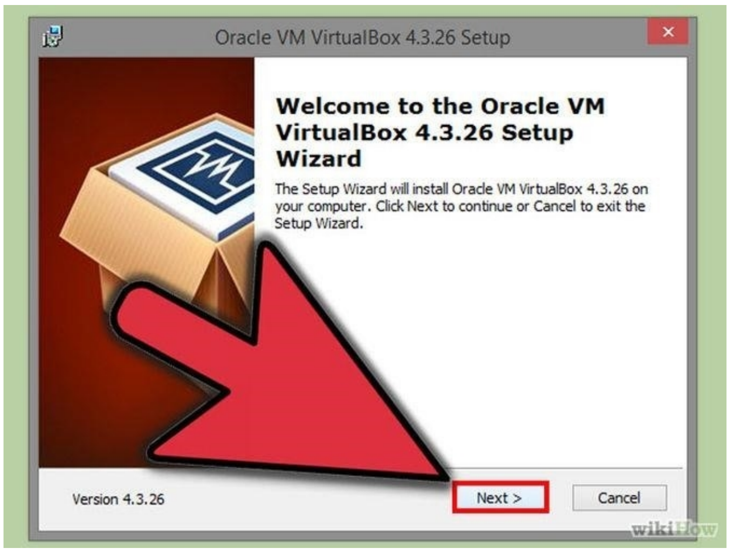

Oracle VM VirtualBox 4.3.26 Setup
Welcome to the Oracle VM VirtualBox 4.3.26 Setup Wizard
The Setup Wizard will install Oracle VM VirtualBox 4.3.26 on your computer. Click Next to continue or Cancel to exit the Setup Wizard.
Version 4.3.26
Next >
Cancel
wikiHow

Fig 2.2

STEP 2:- INSTALL VIRTUAL BOX PROGRAM

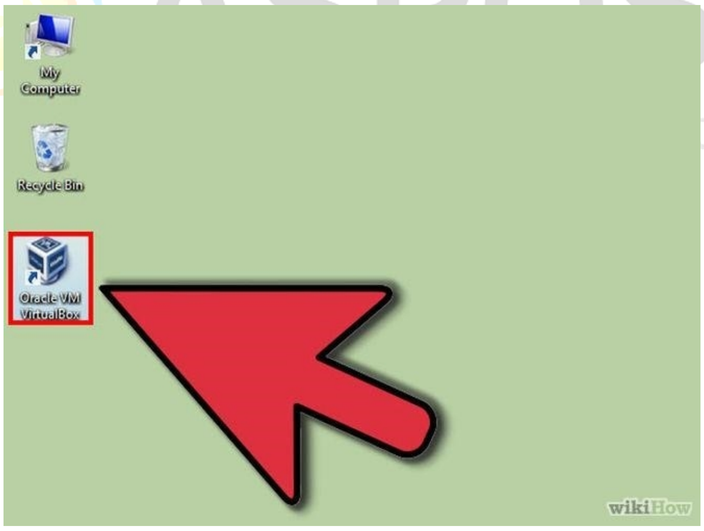


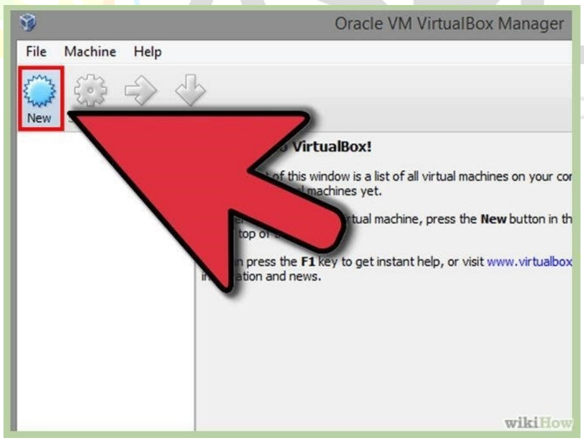
Fig 2.3

STEP 3:- START THE PROGRAM.

Fig 2.4

➢ **IF WE WANT TO ADD NEW OPERATING SYSTEM THEN FOR THAT STEPS ARE:-**

STEP 1:- Click the new button , this will open the wizard that will guide you through which process to create new virtual machine.



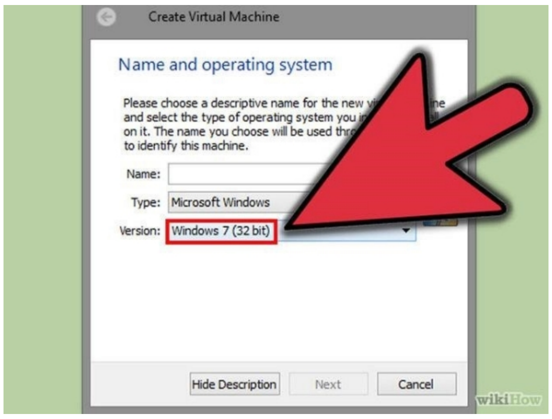
of this window is a list of all virtual machines on your co
machines yet.

tual machine, press the **New** button in th
top of

n press the **F1** key to get instant help, or visit www.virtualbox
ation and news.

Fig 8.5

STEP 2:- identify operating system.



Name and operating system

Please choose a descriptive name for the new v ne and select the type of operating system you i on it. The name you choose will be used thr to identify this machine.

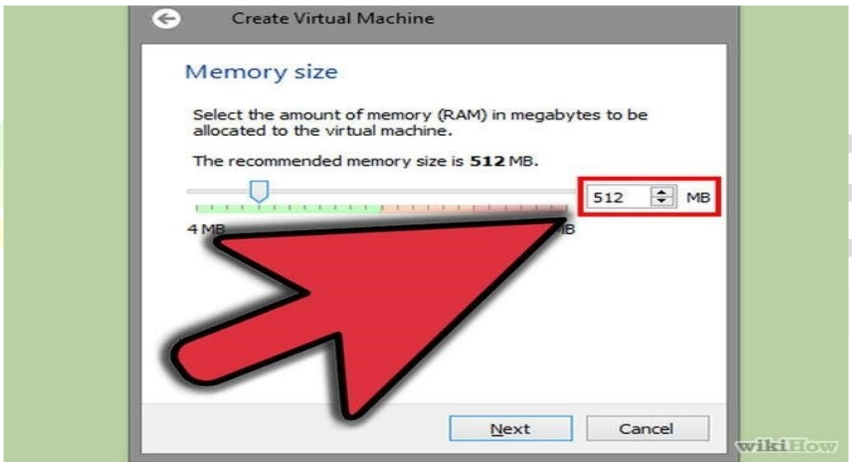
Name:

Type: Microsoft Windows

Version: Windows 7 (32 bit)

Fig 8.6

Step 3:- Set out the RAM.



Memory size

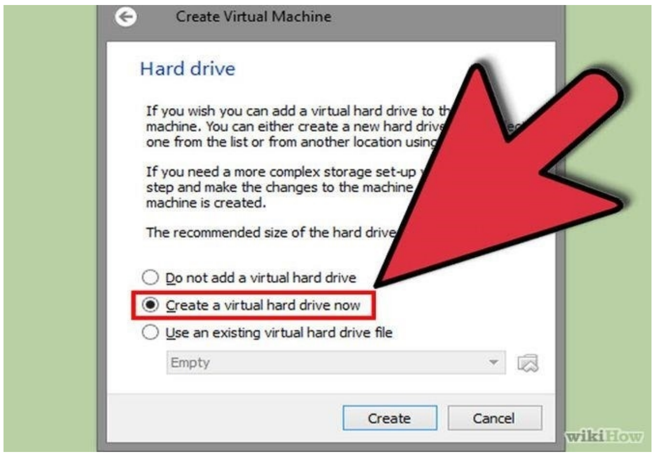
Select the amount of memory (RAM) in megabytes to be allocated to the virtual machine.

The recommended memory size is **512** MB.

512 MB

Fig 8.7

Step 4:- Create a virtual hard drive



Hard drive

If you wish you can add a virtual hard drive to th machine. You can either create a new hard driv one from the list or from another location using

If you need a more complex storage set-up step and make the changes to the machine machine is created.

The recommended size of the hard drive

Do not add a virtual hard drive [✓ Create a virtual hard drive now] Use an existing virtual hard drive file

Empty

Fig 8.8

Step 5:- Start operating system installation.

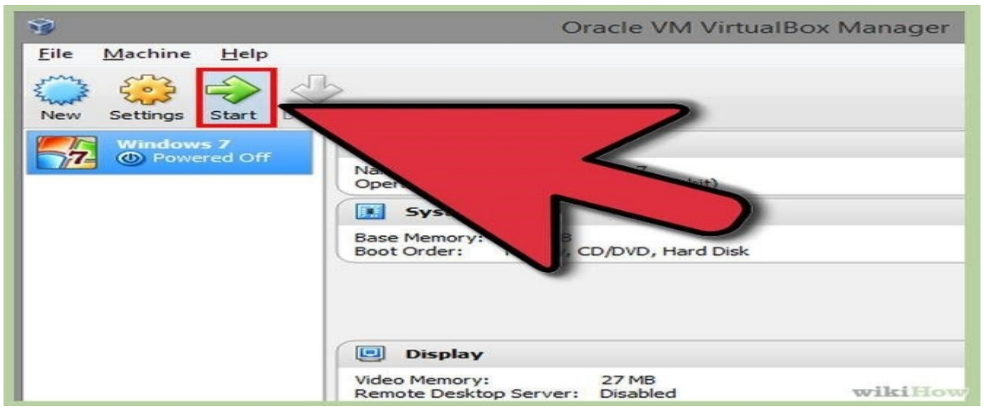
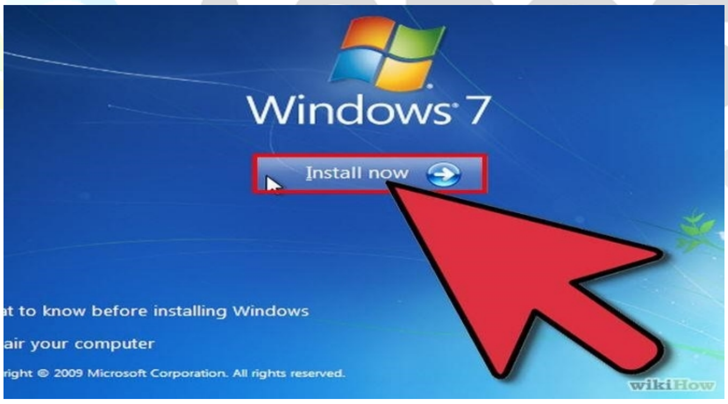


Fig 8.9

Step 6:- INSTALLING THE OPERATING SYSTEM


Windows 7
Install now
t to know before installing Windows
air your computer
right © 2009 Microsoft Corporation. All rights reserved.
wikiHow