

Project on Operating system.....



HONOURABLE FACULTY---

Mam Shivali Chopra.



Student Details

Name-- SHUBHAM PANDEY

Reg No-- 12221997.

Section-- K22 SW.

Roll NO- - 05.

Subject-- OPERATING SYSTEM.



Acknowledgement.....

First of all , I would like to give special thanks to mam shivali chopra , to provide the opportunity to try to do some practical project in field of the operating system.

This project helped me to gain some more knowledge in the field of operating system and to convert a problem statement of operating system in programming code.



PROBLEM STATEMENT.....

Researchers designed one system that classified interactive and noninteractive processes automatically by looking at the amount of terminal I/O. If a process did not input or output to the terminal in a 1-second interval, the process was classified as noninteractive and was moved to a lower-priority queue. In response to this policy, one programmer modified his programs to write an arbitrary character to the terminal at regular intervals of less than 1 second. The system gave his programs a high priority, even though the terminal output was completely meaningless.



DESCRIPTION.....

In operating systems, processes are often classified as interactive or non-interactive based on their behavior and how they interact with the system and users. These classifications are used to help the operating system allocate system resources and determine scheduling priorities.....



Interactive Processes.....

User Interaction: Interactive processes are those that are typically initiated by a user and require user input or provide user feedback. These processes are associated with programs that run in the foreground and directly interact with the user.

Responsiveness: Interactive processes need to be responsive to user actions and should provide real-time or near-real-time feedback. They are given higher scheduling priorities to ensure that the user's actions are quickly addressed.

Example- Graphical user interfaces, text editors, command line interfaces etc.



Non-Interactive Processes

Background Tasks:- Non-interactive processes are background tasks that typically do not require direct user interaction. They may run in the background, performing tasks such as system maintenance, data processing, or batch jobs.

Low Priority:- Non-interactive processes are given lower scheduling priorities compared to interactive processes. This is because they don't require the same level of responsiveness and should not disrupt the user's experience.

Example- system backup , Database management maintain etc.



Summary and Use of it.....

The distinction between interactive and non-interactive processes is essential for the operating system's scheduler to efficiently manage system resources. Interactive processes are given priority to ensure a smooth user experience, while non-interactive processes are allowed to run in the background without hindering the system's responsiveness to user-initiated tasks.



code----

```
/*
```

```
*****CODE WRITTEN BY*****
```

```
NAME- SHUBHAM PANDEY
```

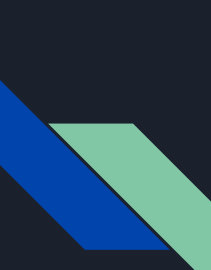
```
SECTION- K22SW
```

```
ROLL.NO= 5
```

```
REG NO= 12221997
```

```
SUBJECT- O.S
```

```
* /
```



```
# include<iostream>
# include<bits/stdc++.h>
using namespace std;
int main(){
    //taking a variable for no of proceeses to take input.

    int pno;

    //taking input from the user
    cout<<"Enter the No of process you want: ";
    cin>>pno;

    cout<<"enter the Time in milliseconds your processes....."<<endl;

    //making a vector to store the flag of given value either 0 or 1.....
    vector<int>v;
```



```
//making a vector to store the values of diffrent processes.
```

```
vector<int>v1;
```

```
//using loop to iterate till the number of process to take input
```

```
for(int i=0;i<pno;i++){
```

```
    cout<<"enter the time for Process P" <<i<<" : ";
```

```
    int x;
```

```
    cin>>x;
```

```
    v1.push_back(x);
```

```
    if(x<=1000){
```

```
        v.push_back(1);
```

```
    }
```

```
    else{
```

```
        v.push_back(0);
```

```
    }
```

```
}
```



```
//displaying all the given credentials>>>>>>>
```

```
cout<<"*****\n"
```

```
cout<<"Processes\t\t"<<"Time\t\t\t"<<"Status\t\t"<<endl;
```

```
cout<<"*****\n"
```

```
for(int i=0;i<pno;i++){
```

```
    if(v[i]==1){
```

```
        cout<<" P"<<i<<": \t\t\t"<<v1[i]<<"\t\t\t"<<"Interactive"<<endl;
```

```
    }
```

```
    else{
```

```
        cout<<" P"<<i<<": \t\t\t"<<v1[i]<<"\t\t\t"<<"Non-Interactive"<<endl;
```

```
    }
```

```
}
```

```
return 0;
```

```
}
```



Test Cases-----

If you are adding time greater than 1000 milliseconds it will become non-interactive else it will remain interactive process.

1. 234
2. 678
3. 1003
4. 456
5. 1233



6. 45521

7. 100

8. 999

9. 1000

10. 4200

The results are displayed on the screen on the basis of the interactive and non- interactive processes according to the specified conditional statement of 1 sec.

snapshots

```
C:\WINDOWS\system32\cmd. X + v
(c) Microsoft Corporation. All rights reserved.

C:\Users\lionp>cd C:\Users\lionp\OneDrive\Desktop\os project

C:\Users\lionp\OneDrive\Desktop\os project>g++ -o newfile Os_project.cpp

C:\Users\lionp\OneDrive\Desktop\os project>Os_project
Enter the No of process you want: 10
enter the Time in milliseconds your processes.....
enter the time for Process P0 : 234
enter the time for Process P1 : 678
enter the time for Process P2 : 1003
enter the time for Process P3 : 456
enter the time for Process P4 : 1233
enter the time for Process P5 : 45521
enter the time for Process P6 : 100
enter the time for Process P7 : 999
enter the time for Process P8 : 1000
enter the time for Process P9 : 4200
*****
Processes          Time          Status
*****
P0:                234          Interactive
P1:                678          Interactive
P2:               1003        Non-Interactive
P3:                456          Interactive
P4:               1233        Non-Interactive
P5:             45521        Non-Interactive
P6:                100          Interactive
P7:                999          Interactive
P8:              1000          Interactive
P9:              4200        Non-Interactive

C:\Users\lionp\OneDrive\Desktop\os project>
```



Complexity and Algorithm used>>>>

Time complexity-- $O(n)$. n =input size

Space Complexity-- $O(n)$. n =input size

ALGORITHM-- Language used--- (c++).

- ❑ Used the concept of vectors (dynamic arrays).
- ❑ Traversing of vectors.
- ❑ Used For loop as a tool.