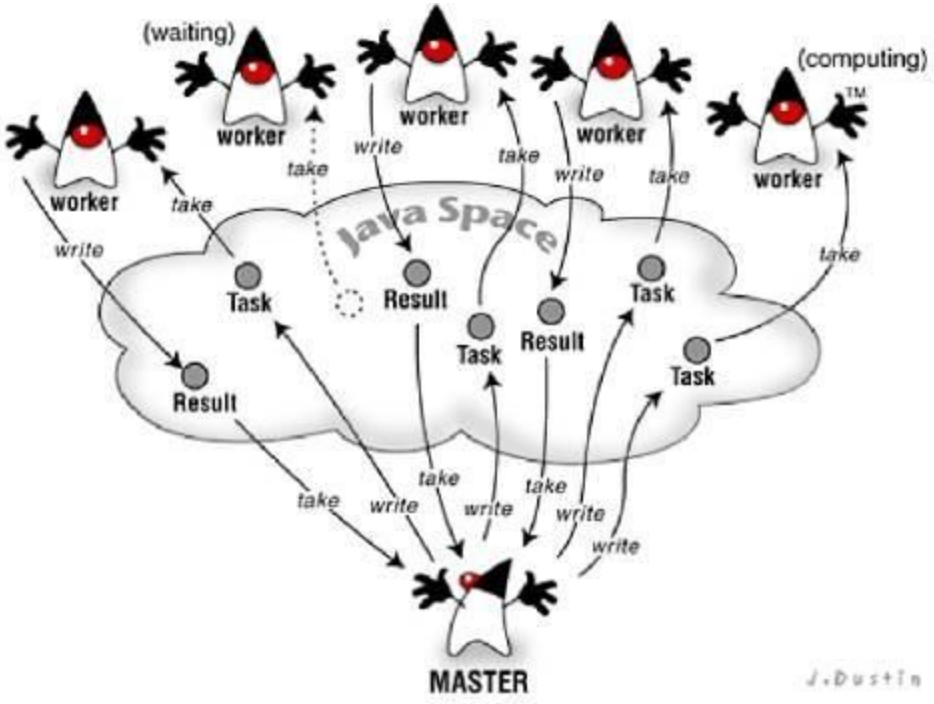
Lab Exercise 5
Master -Worker Search Engine

Name: Ayush Sharma
Reg. No: 15BCE1335
Faculty: Prof. Gayatri. R

1. A search engine can be implemented using a farm of servers; each contains a subset of data that can be searched. Assume that this farm server has a single front-end that interacts with clients who submit queries. Implement the above server form using master-worker pattern.



The Master-Worker Problem is used for parallel processing. It follows a simple approach that allows applications to perform simultaneous processing across multiple machine or processes.

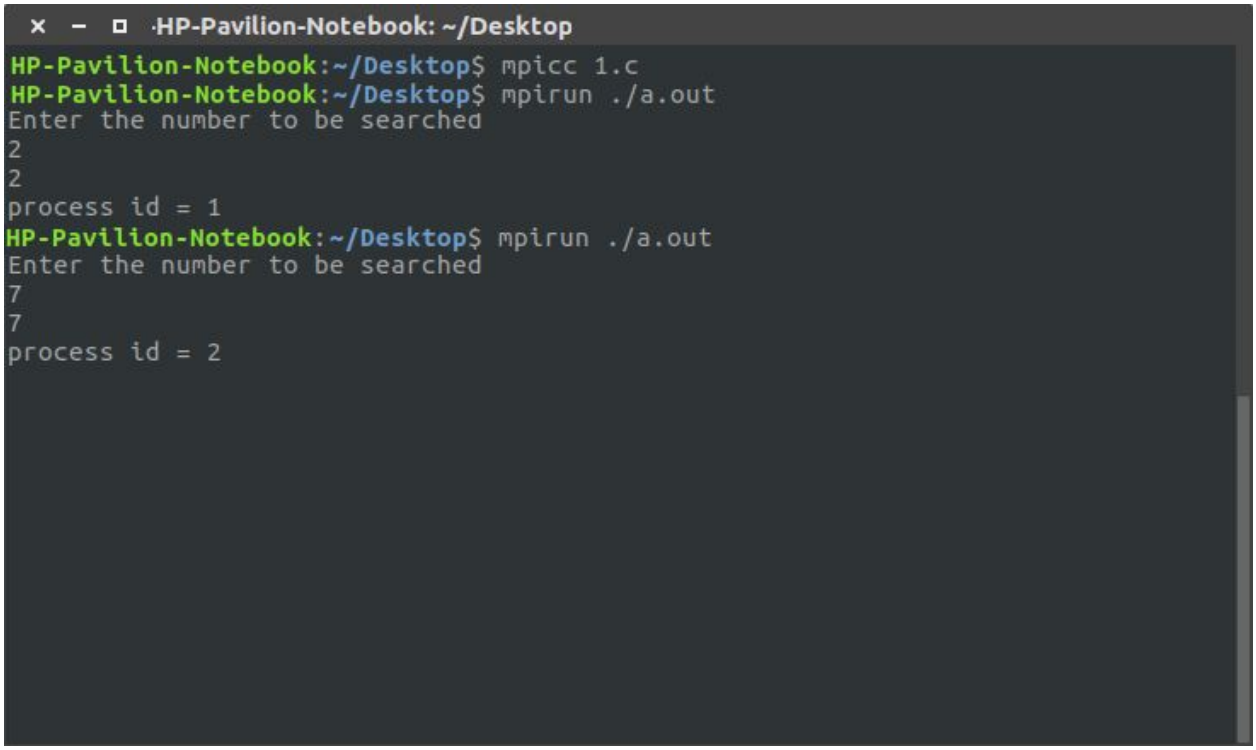
Here in the code:
I. Process=0 is **master**
II. Process=1 or greater are **worker**

Code:

```
#include <stdio.h>
#include <mpi.h>

int main(int argc, char** argv) {
  int n , i  , match = 0 , process = 0 , id = -1, numWorkers = -1,
length = -1;
  char hostName[MPI_MAX_PROCESSOR_NAME];
  int a[10]={2,4,6,8,10,12,14,16,18,20};
  int b[10]={1,3,5,7,9,11,13,15,17,19};
  MPI_Init(&argc, &argv);
  MPI_Comm_rank(MPI_COMM_WORLD, &id);
  MPI_Comm_size(MPI_COMM_WORLD, &numWorkers);
  MPI_Get_processor_name (hostName, &length);
  if ( id == 0 )
  {
    printf("Enter the number to be searched\n");
    scanf("%d",&n);
    for(i=0;i<10;i++)
      {
      if(a[i]==n)
      {
        match = 1;
        process = 1;
      }
      }
    for(i=0;i<10;i++)
      {
      if(b[i]==n)
      {
        match = 1;
        process = 2;
      }
      }
    printf("%d\n",n);
  MPI_Finalize();
  return 0;
}
```

Screenshot:


HP-Pavilion-Notebook: ~/Desktop
HP-Pavilion-Notebook:~/Desktop$ mpicc 1.c
HP-Pavilion-Notebook:~/Desktop$ mpirun ./a.out
Enter the number to be searched
2
2
process id = 1
HP-Pavilion-Notebook:~/Desktop$ mpirun ./a.out
Enter the number to be searched
7
7
process id = 2