```
Script started on 2021-09-15 23:01:10-04:00
[TERM="xterm-256color" TTY="/dev/pts/1" COLUMNS="135"
LINES="35"]
#]0;jmw75@gold28:
~/cs374/Project02##[01;32mjmw75@gold28#[00m:#[01;34m~/c
s374/Project02#[00m$ cat mW#orker.c
/* Project02
 * Assignment Name: mWorker.c
 * ... Assignment to give practice using master-worker
and message-passing patterns ...
 * Joel Adams, Calvin College, September 2013.
 * Modified by: John White, Calvin University, 2021
 * Date: September 15, 2021
 */

#include <stdio.h>   // printf()
#include <mpi.h>     // MPI
#include <stdlib.h>  // malloc()
#include <string.h>  // strlen()

int odd(int number) { return number % 2; }

int main(int argc, char** argv) {
    int id = -1, numProcesses = -1, length = -1;
    char * sendString = NULL;
    char * receivedString = NULL;
    char hostName[MPI_MAX_PROCESSOR_NAME];
    MPI_Status status;
    const int SIZE = (32+MPI_MAX_PROCESSOR_NAME) *
sizeof(char);

    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &id);
    MPI_Comm_size(MPI_COMM_WORLD, &numProcesses);
    MPI_Get_processor_name (hostName, &length);

    double startTime = 0.0, totalTime = 0.0;
    startTime = MPI_Wtime();

    if (numProcesses > 1 && !odd(numProcesses) ) {
        sendString = (char*) malloc( SIZE );
        receivedString = (char*) malloc( SIZE );
        sprintf(sendString, "%d ", id);

        if ( id == 0 ) {  // odd processes send, then
receive
            MPI_Send(sendString, strlen(sendString) +
1, MPI_CHAR, id+1, id, MPI_COMM_WORLD);
            MPI_Recv(receivedString, SIZE, MPI_CHAR,
numProcesses - 1, numProcesses - 1,
                       MPI_COMM_WORLD, &status);
            totalTime = MPI_Wtime() - startTime;
            printf("%s ", receivedString);
            printf("\ntime: %f secs\n", totalTime);
        } else {          // even processes receive,
then send 22
            MPI_Recv(receivedString, SIZE, MPI_CHAR,
id-1, id-1,
                       MPI_COMM_WORLD, &status);
            printf("%s ", receivedString);
            MPI_Send(sendString, strlen(sendString) +
1, MPI_CHAR, (id+1) % numProcesses, id,
MPI_COMM_WORLD);
        }

    } else if ( !id) {  // only process 0 does this
part
        printf("\nPlease run this program using -np N
where N is positive and even.\n\n");
    }

    if ( id == numProcesses - 1) {
        //printf("%c", '\n');
        //printf("\nTime: %f secs\n", totalTime);
    }

    free(sendString); free(receivedString);
    MPI_Finalize();
    return 0;
}
#]0;jmw75@gold28:
~/cs374/Project02##[01;32mjmw75@gold28#[00m:#[01;34m~/c
s374/Project02#[00m$ cat mWorker.c ##############mpirun
-np 8 -machinefile ./hosts
./mWorker##########################################[7Pcc
mWorker.c -Wall -o mWorker -lm
#]0;jmw75@gold28:
~/cs374/Project02##[01;32mjmw75@gold28#[00m:#[01;34m~/c
s374/Project02#[00m$ mpicc mWorker.c -Wall -o mWorker -
lm#####################################[22Pcat
mWorker.c ##############mpirun -np 8 -machinefile
./hosts ./mWorker
0  1  2  3  4  5  6  7
time: 0.006685 secs
#]0;jmw75@gold28:
~/cs374/Project02##[01;32mjmw75@gold28#[00m:#[01;34m~/c
s374/Project02#[00m$ exit

Script done on 2021-09-15 23:01:40-04:00
[COMMAND_EXIT_CODE="0"]
```