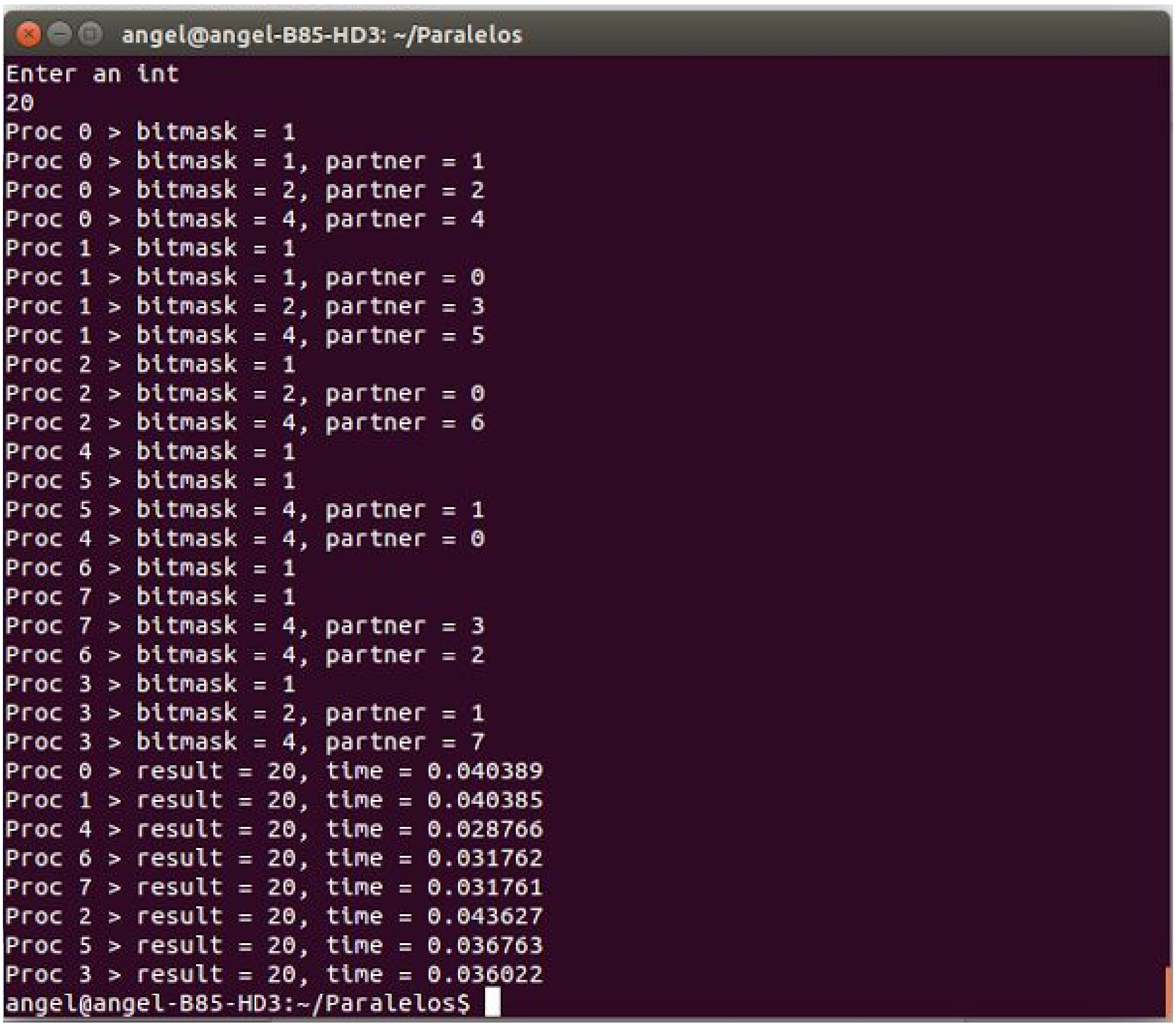
IMPLEMENTACIÓN DE LAS FUNCIONES Bcast y Reduce DE MPI

Bcast:

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

int Bcast(int in_val, int my_rank, int p, MPI_Comm comm);

int main(int argc, char* argv[]) {
   int p, my_rank;
   MPI_Comm comm;
   int result, in_val;
   double my_start, my_finish, my_elapsed;

   MPI_Init(&argc, &argv);
   comm = MPI_COMM_WORLD;
   MPI_Comm_size(comm, &p);
   MPI_Comm_rank(comm, &my_rank);

   if (my_rank == 0) {
      printf("Enter an int\n");
      scanf("%d", &in_val);
   }
   MPI_Barrier(MPI_COMM_WORLD);
   my_start=MPI_Wtime();
   result = Bcast(in_val, my_rank, p, comm);
   //MPI_Bcast(&in_val, 1, MPI_INT, 0, MPI_COMM_WORLD);
   MPI_Barrier(MPI_COMM_WORLD);
   my_finish=MPI_Wtime();
   my_elapsed=my_finish-my_start;
   printf("Proc %d > result = %d, time = %f\n", my_rank, result, my_elapsed);
   //printf("Proc %d > result = %d, time = %f\n", my_rank, in_val,
my_elapsed);

   MPI_Finalize();
   return 0;
}  /* main */

/*-----------------------------------------------------------------*/
/* Function:    Bcast
 *          0 000 001 010 100
 *          1 001 000 011 101
 *          2 010  x  000 110
 *          3 011  x  001 111
 *          4 100  x   x  000
 *          5 101  x   x  001
 *          6 110  x   x  010
 *          7 111  x   x  011
 */
int Bcast(int in_val, int my_rank, int p, MPI_Comm comm) {
   int        partner;
   unsigned   bitmask = 1;
   int        participate = bitmask << 1;

//#  ifdef DEBUG
   printf("Proc %d > bitmask = %d\n", my_rank, bitmask);
   fflush(stdout);
//#  endif
   while (bitmask < p) {
      if (my_rank < participate) {
         partner = my_rank ^ bitmask;
//#        ifdef DEBUG
         printf("Proc %d > bitmask = %d, partner = %d\n",
            my_rank, bitmask, partner);
         fflush(stdout);
//#        endif
         if (my_rank < partner) {
            if (partner < p )
               MPI_Send(&in_val, 1, MPI_INT, partner, 0, comm);
         } else {
            MPI_Recv(&in_val, 1, MPI_INT, partner, 0, comm,
                  MPI_STATUS_IGNORE);
         }
      }
      bitmask <<= 1;
      participate <<= 1;
   }
   return in_val;

}  /* Bcast */
```

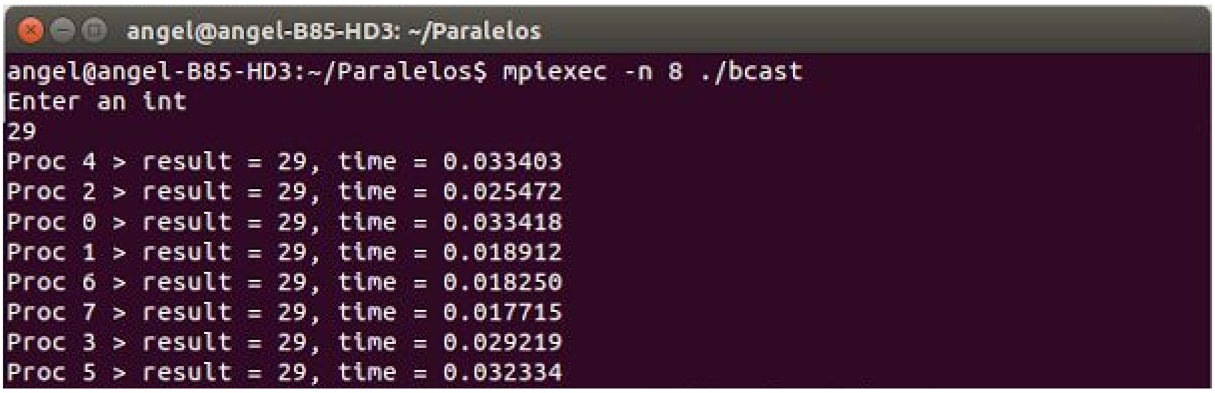
Ejecución de la implementación con 8 procesos:



```
Enter an int
20
Proc 0 > bitmask = 1
Proc 0 > bitmask = 1, partner = 1
Proc 0 > bitmask = 2, partner = 2
Proc 0 > bitmask = 4, partner = 4
Proc 1 > bitmask = 1
Proc 1 > bitmask = 1, partner = 0
Proc 1 > bitmask = 2, partner = 3
Proc 1 > bitmask = 4, partner = 5
Proc 2 > bitmask = 1
Proc 2 > bitmask = 2, partner = 0
Proc 2 > bitmask = 4, partner = 6
Proc 4 > bitmask = 1
Proc 5 > bitmask = 1
Proc 5 > bitmask = 4, partner = 1
Proc 4 > bitmask = 4, partner = 0
Proc 6 > bitmask = 1
Proc 7 > bitmask = 1
Proc 7 > bitmask = 4, partner = 3
Proc 6 > bitmask = 4, partner = 2
Proc 3 > bitmask = 1
Proc 3 > bitmask = 2, partner = 1
Proc 3 > bitmask = 4, partner = 7
Proc 0 > result = 20, time = 0.040389
Proc 1 > result = 20, time = 0.040385
Proc 4 > result = 20, time = 0.028766
Proc 6 > result = 20, time = 0.031762
Proc 7 > result = 20, time = 0.031761
Proc 2 > result = 20, time = 0.043627
Proc 5 > result = 20, time = 0.036763
Proc 3 > result = 20, time = 0.036022
angel@angel-B85-HD3:~/Paralelos$
```

El tiempo máximo es: 0.043627

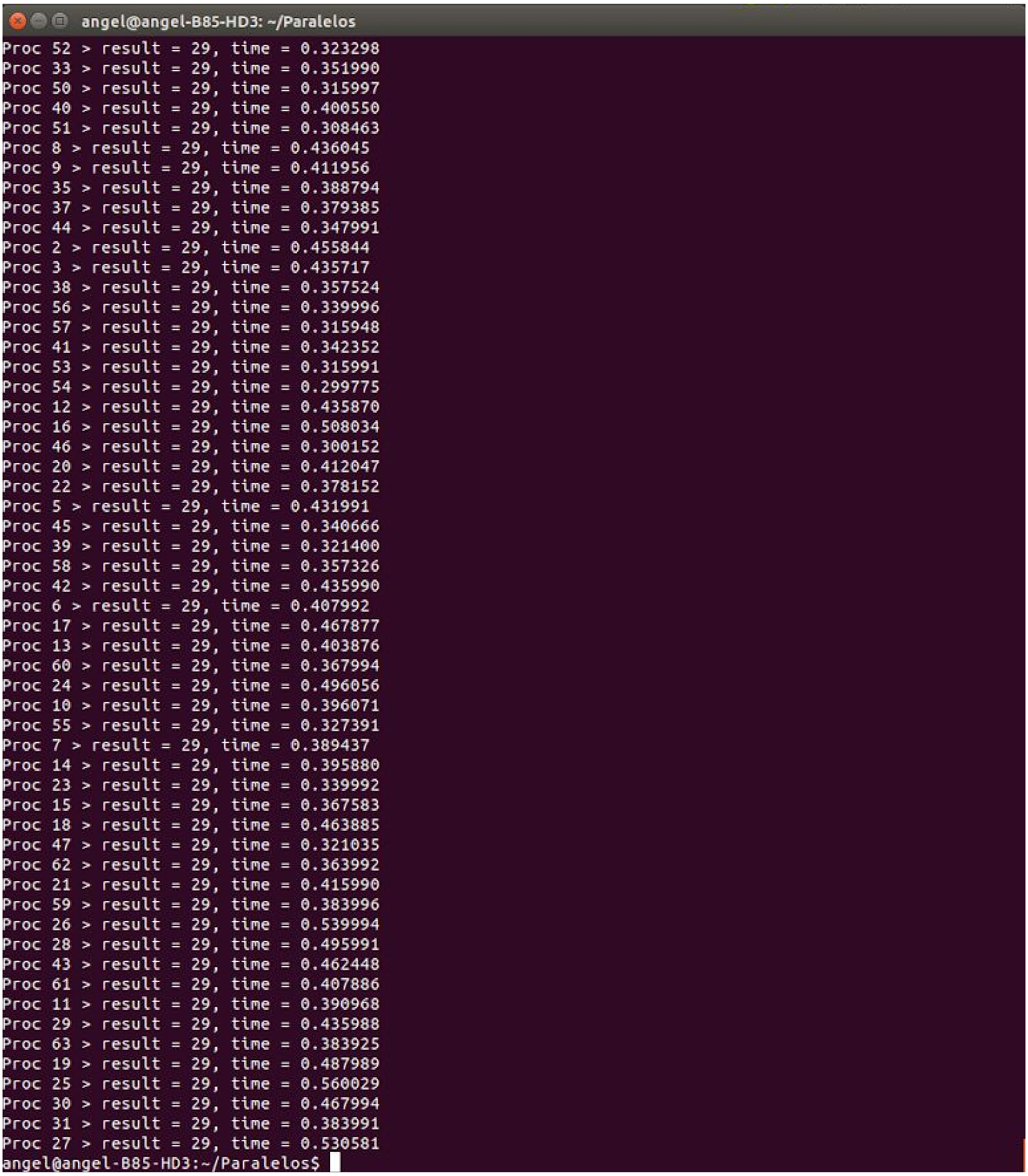
Ejecución de MPI_Bcast con 8 procesos:



```
angel@angel-B85-HD3:~/Paralelos$ mpiexec -n 8 ./bcast
Enter an int
29
Proc 4 > result = 29, time = 0.033403
Proc 2 > result = 29, time = 0.025472
Proc 0 > result = 29, time = 0.033418
Proc 1 > result = 29, time = 0.018912
Proc 6 > result = 29, time = 0.018250
Proc 7 > result = 29, time = 0.017715
Proc 3 > result = 29, time = 0.029219
Proc 5 > result = 29, time = 0.032334
```

El tiempo máximo es: 0.032334

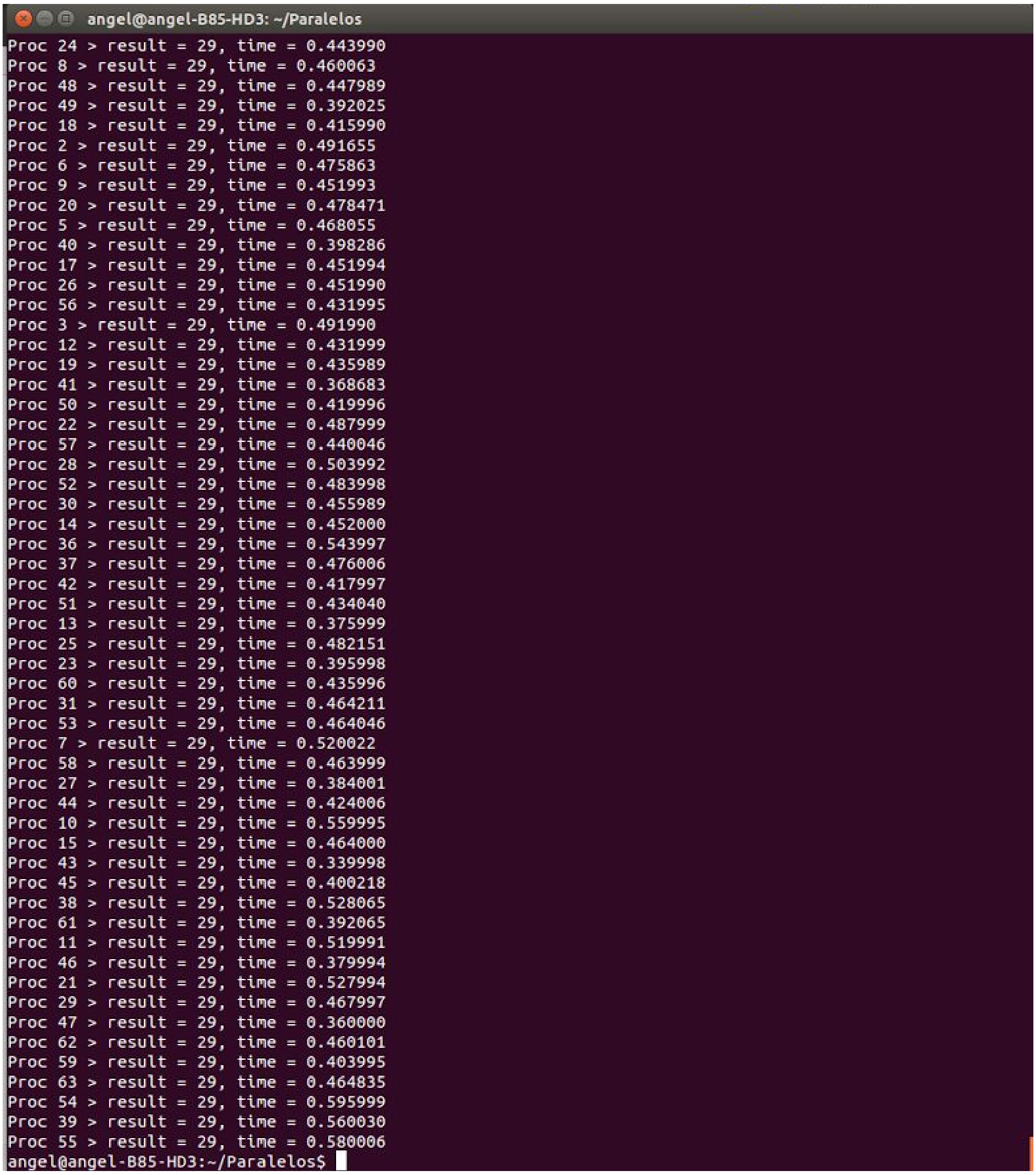
Ejecución de la implementación con 64 procesos:



```
Proc 52 > result = 29, time = 0.323298
Proc 33 > result = 29, time = 0.351990
Proc 50 > result = 29, time = 0.315997
Proc 40 > result = 29, time = 0.400550
Proc 51 > result = 29, time = 0.308463
Proc 8 > result = 29, time = 0.436045
Proc 9 > result = 29, time = 0.411956
Proc 35 > result = 29, time = 0.388794
Proc 37 > result = 29, time = 0.379385
Proc 44 > result = 29, time = 0.347991
Proc 2 > result = 29, time = 0.455844
Proc 3 > result = 29, time = 0.435717
Proc 38 > result = 29, time = 0.357524
Proc 56 > result = 29, time = 0.339996
Proc 57 > result = 29, time = 0.315948
Proc 41 > result = 29, time = 0.342352
Proc 53 > result = 29, time = 0.315991
Proc 54 > result = 29, time = 0.299775
Proc 12 > result = 29, time = 0.435870
Proc 16 > result = 29, time = 0.508034
Proc 46 > result = 29, time = 0.300152
Proc 20 > result = 29, time = 0.412047
Proc 22 > result = 29, time = 0.378152
Proc 5 > result = 29, time = 0.431991
Proc 45 > result = 29, time = 0.340666
Proc 39 > result = 29, time = 0.321400
Proc 58 > result = 29, time = 0.357326
Proc 42 > result = 29, time = 0.435990
Proc 6 > result = 29, time = 0.407992
Proc 17 > result = 29, time = 0.467877
Proc 13 > result = 29, time = 0.403876
Proc 60 > result = 29, time = 0.367994
Proc 24 > result = 29, time = 0.496056
Proc 10 > result = 29, time = 0.396071
Proc 55 > result = 29, time = 0.327391
Proc 7 > result = 29, time = 0.389437
Proc 14 > result = 29, time = 0.395880
Proc 23 > result = 29, time = 0.339992
Proc 15 > result = 29, time = 0.367583
Proc 18 > result = 29, time = 0.463885
Proc 47 > result = 29, time = 0.321035
Proc 62 > result = 29, time = 0.363992
Proc 21 > result = 29, time = 0.415990
Proc 59 > result = 29, time = 0.383996
Proc 26 > result = 29, time = 0.539994
Proc 28 > result = 29, time = 0.495991
Proc 43 > result = 29, time = 0.462448
Proc 61 > result = 29, time = 0.407886
Proc 11 > result = 29, time = 0.390968
Proc 29 > result = 29, time = 0.435988
Proc 63 > result = 29, time = 0.383925
Proc 19 > result = 29, time = 0.487989
Proc 25 > result = 29, time = 0.560029
Proc 30 > result = 29, time = 0.467994
Proc 31 > result = 29, time = 0.383991
Proc 27 > result = 29, time = 0.530581
angel@angel-B85-HD3:~/Paralelos$
```

El tiempo máximo es: 0.530581

Ejecución de MPI_Bcast con 64 procesos:



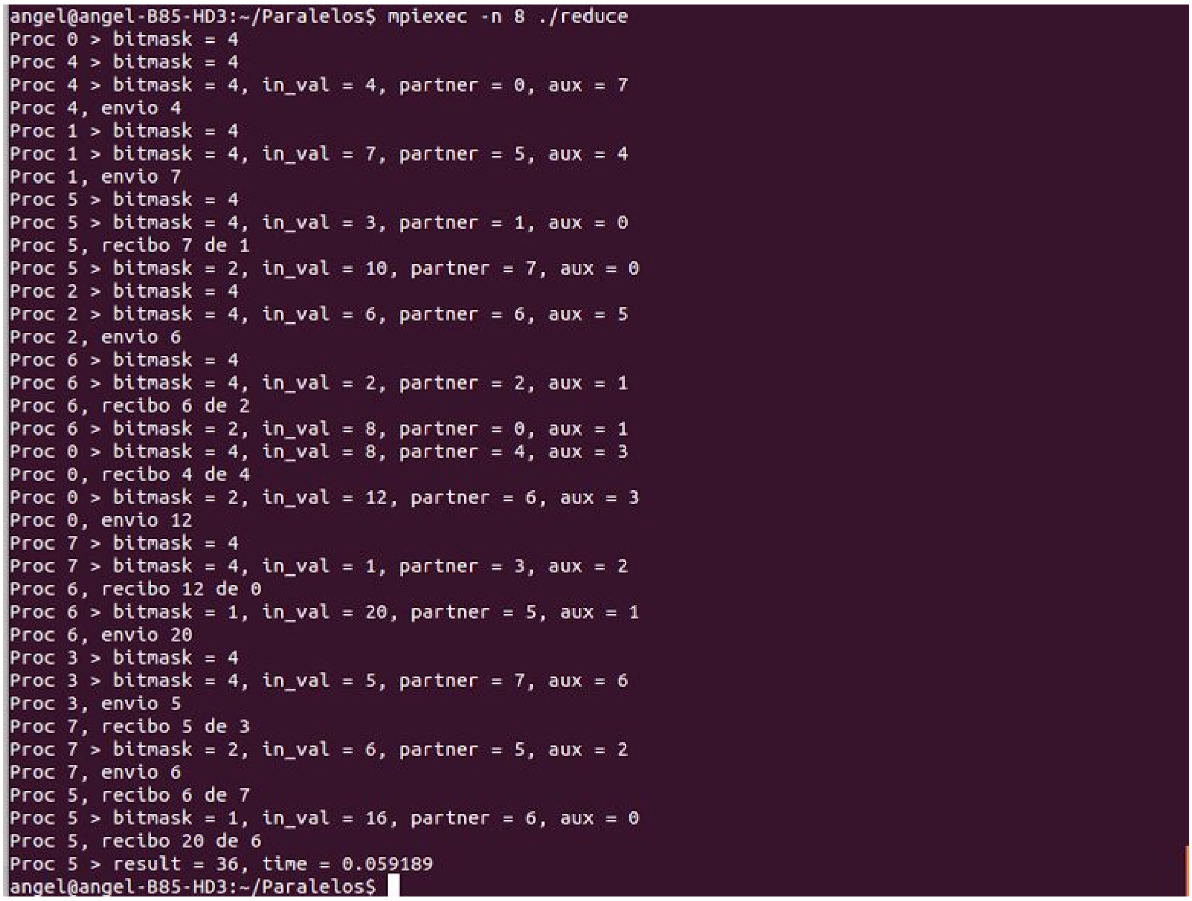
```
Proc 24 > result = 29, time = 0.443990
Proc 8 > result = 29, time = 0.460063
Proc 48 > result = 29, time = 0.447989
Proc 49 > result = 29, time = 0.392025
Proc 18 > result = 29, time = 0.415990
Proc 2 > result = 29, time = 0.491655
Proc 6 > result = 29, time = 0.475863
Proc 9 > result = 29, time = 0.451993
Proc 20 > result = 29, time = 0.478471
Proc 5 > result = 29, time = 0.468055
Proc 40 > result = 29, time = 0.398286
Proc 17 > result = 29, time = 0.451994
Proc 26 > result = 29, time = 0.451990
Proc 56 > result = 29, time = 0.431995
Proc 3 > result = 29, time = 0.491990
Proc 12 > result = 29, time = 0.431999
Proc 19 > result = 29, time = 0.435989
Proc 41 > result = 29, time = 0.368683
Proc 50 > result = 29, time = 0.419996
Proc 22 > result = 29, time = 0.487999
Proc 57 > result = 29, time = 0.440046
Proc 28 > result = 29, time = 0.503992
Proc 52 > result = 29, time = 0.483998
Proc 30 > result = 29, time = 0.455989
Proc 14 > result = 29, time = 0.452000
Proc 36 > result = 29, time = 0.543997
Proc 37 > result = 29, time = 0.476006
Proc 42 > result = 29, time = 0.417997
Proc 51 > result = 29, time = 0.434040
Proc 13 > result = 29, time = 0.375999
Proc 25 > result = 29, time = 0.482151
Proc 23 > result = 29, time = 0.395998
Proc 60 > result = 29, time = 0.435996
Proc 31 > result = 29, time = 0.464211
Proc 53 > result = 29, time = 0.464046
Proc 7 > result = 29, time = 0.520022
Proc 58 > result = 29, time = 0.463999
Proc 27 > result = 29, time = 0.384001
Proc 44 > result = 29, time = 0.424006
Proc 10 > result = 29, time = 0.559995
Proc 15 > result = 29, time = 0.464000
Proc 43 > result = 29, time = 0.339998
Proc 45 > result = 29, time = 0.400218
Proc 38 > result = 29, time = 0.528065
Proc 61 > result = 29, time = 0.392065
Proc 11 > result = 29, time = 0.519991
Proc 46 > result = 29, time = 0.379994
Proc 21 > result = 29, time = 0.527994
Proc 29 > result = 29, time = 0.467997
Proc 47 > result = 29, time = 0.360000
Proc 62 > result = 29, time = 0.460101
Proc 59 > result = 29, time = 0.403995
Proc 63 > result = 29, time = 0.464835
Proc 54 > result = 29, time = 0.595999
Proc 39 > result = 29, time = 0.560030
Proc 55 > result = 29, time = 0.580006
angel@angel-B85-HD3:~/Paralelos$
```

El tiempo máximo es: 0.595999

Reduce:

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

int Reduce(void *my_val, void *in_val, int my_rank, int root, int p, MPI_Comm
comm);

int main(int argc, char* argv[]) {
   int p, my_rank, my_val;
   MPI_Comm comm;
   int result, in_val;

   MPI_Init(&argc, &argv);
   comm = MPI_COMM_WORLD;
   MPI_Comm_size(comm, &p);
   MPI_Comm_rank(comm, &my_rank);

   //shared double global_elapsed;
   double my_start, my_finish, my_elapsed;
      //printf("Proc %d > Enter an int\n", my_rank);
      //scanf("%d", &my_val);
   my_val=p-my_rank;
   in_val=my_val;

   printf("eny %d\n", -1%8);
   MPI_Barrier(MPI_COMM_WORLD);
   my_start=MPI_Wtime();

   result = Reduce(&my_val, &in_val, my_rank, 5, p, comm);
   MPI_Reduce(&my_val, &in_val, 1, MPI_INT, MPI_SUM, 5,
           MPI_COMM_WORLD);
   MPI_Barrier(MPI_COMM_WORLD);
   my_finish=MPI_Wtime();
   my_elapsed=my_finish-my_start;

   //global_elapsed=Global_max(my_elapsed);

   if (my_rank == 5) {
      printf("Proc %d > result = %d, time = %f\n", my_rank, result,
my_elapsed);

   }

   MPI_Finalize();
   return 0;
}  /* main */


/*-----------------------------------------------------------------*/
/* Function:    Reduce
 *          r     001 010 100
 *          -     --- --- ---
 *          0 000 001 010 100
 *          1 001 000 011 101
 *          2 010  x  000 110
 *          3 011  x  001 111
 *          4 100  x   x  000
 *          5 101  x   x  001
 *           6 110  x   x  010
 *           7 111  x   x  011
 */
int Reduce(int my_val, int in_val, int my_rank, int root, int p, MPI_Comm
comm) {
   int        partner, partner1, aux, acum;
   unsigned   bitmask = 1;
   while (bitmask < p) bitmask <<= 1;
   bitmask >>= 1;
   int        participate = bitmask << 1;

//#  ifdef DEBUG
   printf("Proc %d > bitmask = %d\n", my_rank, bitmask);
   fflush(stdout);
//#  endif
   while (bitmask > 0) {
   aux=my_rank-root;
   if (aux<0) aux+=p;
   if (aux < participate) {

         partner = ((aux ^ bitmask)+root)%p;
         partner1 = aux ^ bitmask;
//#        ifdef DEBUG
         printf("Proc %d > bitmask = %d, in_val = %d, partner = %d, aux =
%d\n",
            my_rank, bitmask, in_val, partner,aux);
         fflush(stdout);
//#        endif
         if (aux > partner1) {
            if (partner < p ){
             printf("Proc %d, envio %d\n", my_rank, in_val);
               MPI_Send(&in_val, 1, MPI_INT, partner, 0, comm);
            }
         } else {
            MPI_Recv(&acum, 1, MPI_INT, partner, 0, comm,
                  MPI_STATUS_IGNORE);
            printf("Proc %d, recivo %d de %d\n", my_rank, acum,partner);
            in_val+=acum;

         }
      }
      bitmask >>= 1;
      participate >>= 1;
   }
   return in_val;

}
```

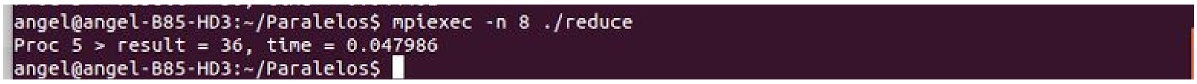
Ejecución de la implementación con 8 procesos:



```
angel@angel-B85-HD3:~/Paralelos$ mpiexec -n 8 ./reduce
Proc 0 > bitmask = 4
Proc 4 > bitmask = 4
Proc 4 > bitmask = 4, in_val = 4, partner = 0, aux = 7
Proc 4, envio 4
Proc 1 > bitmask = 4
Proc 1 > bitmask = 4, in_val = 7, partner = 5, aux = 4
Proc 1, envio 7
Proc 5 > bitmask = 4
Proc 5 > bitmask = 4, in_val = 3, partner = 1, aux = 0
Proc 5, recibo 7 de 1
Proc 5 > bitmask = 2, in_val = 10, partner = 7, aux = 0
Proc 2 > bitmask = 4
Proc 2 > bitmask = 4, in_val = 6, partner = 6, aux = 5
Proc 2, envio 6
Proc 6 > bitmask = 4
Proc 6 > bitmask = 4, in_val = 2, partner = 2, aux = 1
Proc 6, recibo 6 de 2
Proc 6 > bitmask = 2, in_val = 8, partner = 0, aux = 1
Proc 0 > bitmask = 4, in_val = 8, partner = 4, aux = 3
Proc 0, recibo 4 de 4
Proc 0 > bitmask = 2, in_val = 12, partner = 6, aux = 3
Proc 0, envio 12
Proc 7 > bitmask = 4
Proc 7 > bitmask = 4, in_val = 1, partner = 3, aux = 2
Proc 6, recibo 12 de 0
Proc 6 > bitmask = 1, in_val = 20, partner = 5, aux = 1
Proc 6, envio 20
Proc 3 > bitmask = 4
Proc 3 > bitmask = 4, in_val = 5, partner = 7, aux = 6
Proc 3, envio 5
Proc 7, recibo 5 de 3
Proc 7 > bitmask = 2, in_val = 6, partner = 5, aux = 2
Proc 7, envio 6
Proc 5, recibo 6 de 7
Proc 5 > bitmask = 1, in_val = 16, partner = 6, aux = 0
Proc 5, recibo 20 de 6
Proc 5 > result = 36, time = 0.059189
angel@angel-B85-HD3:~/Paralelos$ 
```

El tiempo es: 0.059189

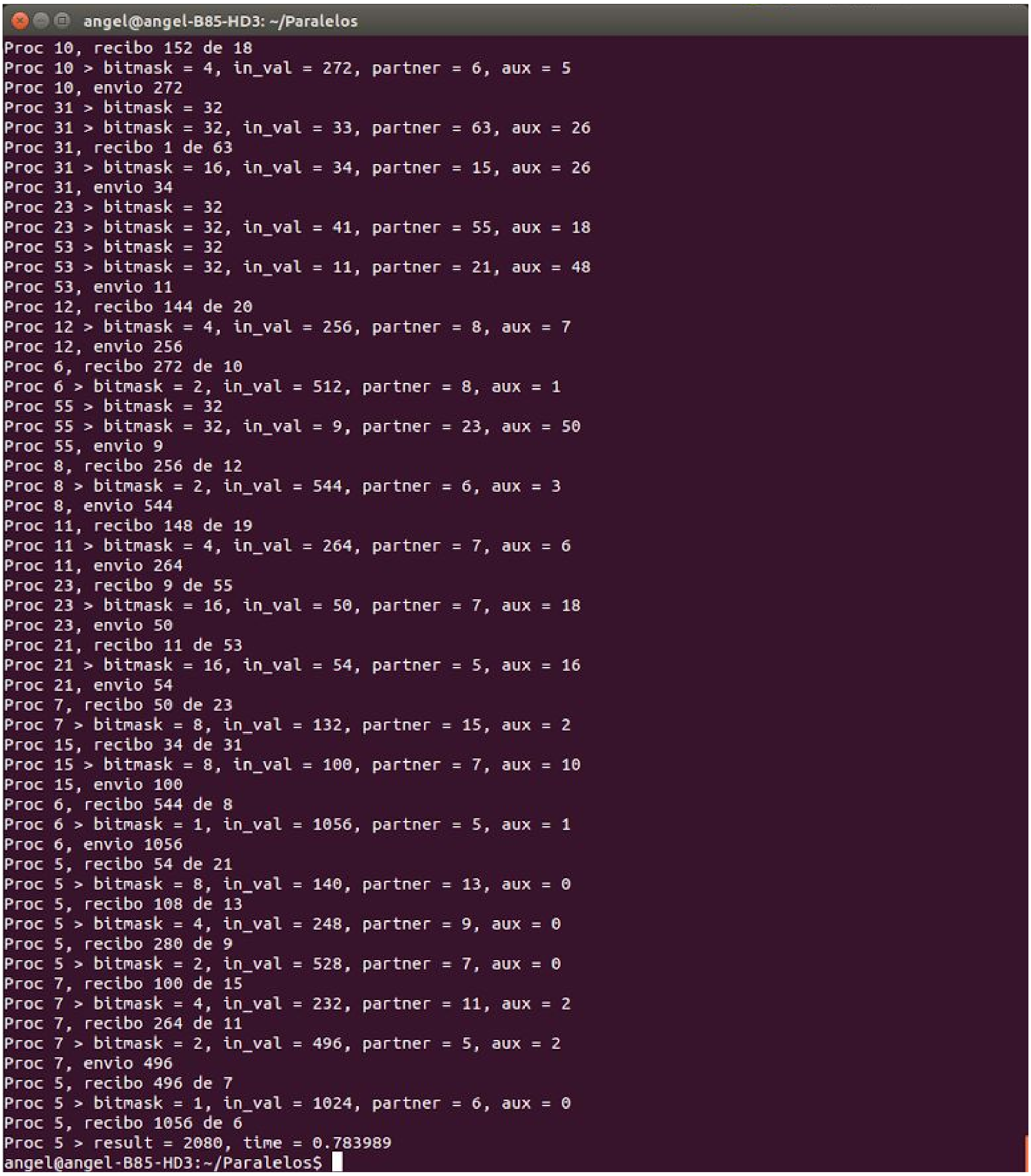
Ejecución de MPI_Reduce con 8 procesos:



```
angel@angel-B85-HD3:~/Paralelos$ mpiexec -n 8 ./reduce
Proc 5 > result = 36, time = 0.047986
angel@angel-B85-HD3:~/Paralelos$ 
```

El tiempo es: 0.047986

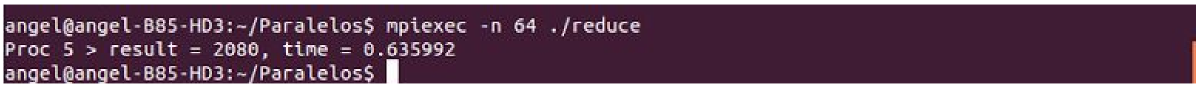
Ejecución de la implementación con 64 procesos:



```
Proc 10, recibo 152 de 18
Proc 10 > bitmask = 4, in_val = 272, partner = 6, aux = 5
Proc 10, envio 272
Proc 31 > bitmask = 32
Proc 31 > bitmask = 32, in_val = 33, partner = 63, aux = 26
Proc 31, recibo 1 de 63
Proc 31 > bitmask = 16, in_val = 34, partner = 15, aux = 26
Proc 31, envio 34
Proc 23 > bitmask = 32
Proc 23 > bitmask = 32, in_val = 41, partner = 55, aux = 18
Proc 53 > bitmask = 32
Proc 53 > bitmask = 32, in_val = 11, partner = 21, aux = 48
Proc 53, envio 11
Proc 12, recibo 144 de 20
Proc 12 > bitmask = 4, in_val = 256, partner = 8, aux = 7
Proc 12, envio 256
Proc 6, recibo 272 de 10
Proc 6 > bitmask = 2, in_val = 512, partner = 8, aux = 1
Proc 55 > bitmask = 32
Proc 55 > bitmask = 32, in_val = 9, partner = 23, aux = 50
Proc 55, envio 9
Proc 8, recibo 256 de 12
Proc 8 > bitmask = 2, in_val = 544, partner = 6, aux = 3
Proc 8, envio 544
Proc 11, recibo 148 de 19
Proc 11 > bitmask = 4, in_val = 264, partner = 7, aux = 6
Proc 11, envio 264
Proc 23, recibo 9 de 55
Proc 23 > bitmask = 16, in_val = 50, partner = 7, aux = 18
Proc 23, envio 50
Proc 21, recibo 11 de 53
Proc 21 > bitmask = 16, in_val = 54, partner = 5, aux = 16
Proc 21, envio 54
Proc 7, recibo 50 de 23
Proc 7 > bitmask = 8, in_val = 132, partner = 15, aux = 2
Proc 15, recibo 34 de 31
Proc 15 > bitmask = 8, in_val = 100, partner = 7, aux = 10
Proc 15, envio 100
Proc 6, recibo 544 de 8
Proc 6 > bitmask = 1, in_val = 1056, partner = 5, aux = 1
Proc 6, envio 1056
Proc 5, recibo 54 de 21
Proc 5 > bitmask = 8, in_val = 140, partner = 13, aux = 0
Proc 5, recibo 108 de 13
Proc 5 > bitmask = 4, in_val = 248, partner = 9, aux = 0
Proc 5, recibo 280 de 9
Proc 5 > bitmask = 2, in_val = 528, partner = 7, aux = 0
Proc 7, recibo 100 de 15
Proc 7 > bitmask = 4, in_val = 232, partner = 11, aux = 2
Proc 7, recibo 264 de 11
Proc 7 > bitmask = 2, in_val = 496, partner = 5, aux = 2
Proc 7, envio 496
Proc 5, recibo 496 de 7
Proc 5 > bitmask = 1, in_val = 1024, partner = 6, aux = 0
Proc 5, recibo 1056 de 6
Proc 5 > result = 2080, time = 0.783989
angel@angel-B85-HD3:~/Paralelos$
```

El tiempo es: 0.783989

Ejecución de MPI_Reduce con 64 procesos:



```
angel@angel-B85-HD3:~/Paralelos$ mpiexec -n 64 ./reduce
Proc 5 > result = 2080, time = 0.635992
angel@angel-B85-HD3:~/Paralelos$
```

El tiempo es: 0.635992