

MPI Basics - Solutions to Exercises

S. Van Criekingen
UPJV / MeCS

www.mecs.u-picardie.fr

January 28, 2015



Solution of exercise 2 (in C)

```
[...]
int sent_msg, recv_msg;
int partner_rank;

MPI_Init(&argc,&argv);
MPI_Comm_size(MPI_COMM_WORLD,&ntasks);
MPI_Comm_rank(MPI_COMM_WORLD,&rank);

partner_rank = ntasks - (rank+1);
printf ("Rank %d has partner rank %d \n", rank, partner_rank);

dest = partner_rank;
sent_msg = rank;
// printf ("Integer sent by rank %d : %d \n", rank, sent_msg);
MPI_Send (&sent_msg,count,MPI_INT,dest,tag,MPI_COMM_WORLD) ;

source = partner_rank;
MPI_Recv (&recv_msg,count,MPI_INT,source,tag,MPI_COMM_WORLD,&status);
printf ("Integer received by rank %d : %d \n", rank, recv_msg);

MPI_Finalize();
}
```

partner1.c

```
> mpirun -np 3 ./a.out
Rank 0 has partner rank 2
Rank 1 has partner rank 1
Rank 2 has partner rank 0
Integer received by rank 1 : 1
Integer received by rank 0 : 2
Integer received by rank 2 : 0
```

Solution of exercise 3 (first step, in C)

```
[...]
int main(int argc, char *argv[])
{
    int i, ntasks, rank, tossResult;

    MPI_Init(&argc,&argv);
    MPI_Comm_size(MPI_COMM_WORLD, &ntasks);
    MPI_Comm_rank(MPI_COMM_WORLD, &rank);

    srand(rank+time(NULL));
    tossResult = (int) ((double)rand() / ((double)RAND_MAX + 1) * 2);

    printf("On rank %d: tossResult=%d \n", rank, tossResult);

    MPI_Finalize();
}
```

coinTossParallel.c

Solution of exercise 3 (second step, in C)

```
[...]
srand(rank+time(NULL));
nTossMax = 10;
nToss = 0;
sum = 1;
while ((sum !=0) && (sum!=ntasks) && nToss < nTossMax)
{
    nToss += 1;
    tossResult = (int) ((double)rand() / ((double)RAND_MAX + 1) * 2);
    MPI_Allreduce(&tossResult, &sum, 1, MPI_INT, MPI_SUM, MPI_COMM_WORLD);
}
if (rank == 0)
{
    if ((sum ==0) || (sum==ntasks))
    { printf("Unanimity achieved after %d toss(es) \n", nToss); }
    else
    { printf("Unanimity not achieved after %d toss(es) \n", nTossMax) ; }
}
MPI_Finalize();
}
```

coinTossSol.c

(NB : also possible with MPI_Reduce)

Solution of exercise 3 (in Fortran)

[...]

```
call date_and_time(values=timeValues)
timeValues(1) = timeValues(1) + rank
call random_seed(size=K)
call random_seed(put=timeValues(1:K))
do while ((sum<ntasks).and.(sum>0).and.(nToss < nTossMax))
  call random_number(random)
  tossResult = nint(random)
  call MPI_AllReduce(tossResult, sum, 1, MPI_INTEGER, MPI_SUM, MPI_COMM_WORLD, ierr)
  nToss = nToss+1
  if (rank==0) write(*,'(a,i2,a,i2)') 'nToss= ', nToss, ' sum = ', sum
end do
if (rank==0) then
  if ((sum==0).or.(sum==ntasks)) then
    write(*,'(a,i2,a)') 'Unanimity achieved after', nToss, ' tosses.'
  else
    write(*,'(a,i2,a)') 'Unanimity not achieved after', nToss, ' tosses.'
  end if
end if
call MPI_FINALIZE(ierr)
end
```

coinTossSol.f90

(NB : also possible with MPI_Reduce)