Aide mémoire MPI en C

Version 4.6 – 01 juin 2015

1 Environnement

1.1 Initialisation de l'environnement MPI

```
int MPI_Init(int * argc, char **argv)
```

1.2 Rang du processus

```
int MPI_Comm_rank(MPI_Comm comm, int * rang)
```

1.3 Nombre de processus

```
int MPI_Comm_size(MPI_Comm comm, int * nb_procs)
```

1.4 Désactivation de l'environnement MPI

```
int MPI_Finalize(void)
```

1.5 Arret d'un programme MPI

```
int MPI_Abort(MPI_Comm comm, int error)
```

1.6 Prise de temps

```
double MPI_Wtime(void)
```

2 Communications point à point

2.1 Envoi de message

```
int MPI_Send(
        const void *    message,
        int             longueur,
        MPI_Datatype    type,
        int             rang_dest,
        int             etiquette,
        MPI_Comm        comm)
```

2.2 Envoi non bloquant de message

```
int MPI_Isend(
        const void *    message,
        int             longueur,
        MPI_Datatype    type,
        int             rang_dest,
        int             etiquette,
        MPI_Comm        comm,
        MPI_Request *   requete)
```

2.3 Reception de message

```
int MPI_Recv(
        void *          message,
        int             longueur,
        MPI_Datatype    type,
        int             rang_source,
        int             etiquette,
        MPI_Comm        comm,
        MPI_Status *    statut)
```

2.4 Reception non bloquant de message

```
int MPI_Irecv(
        void *          message,
        int             longueur,
        MPI_Datatype    type,
        int             rang_source,
        int             etiquette,
        MPI_Comm        comm,
        MPI_Request *   requete)
```

2.5 Envoi et reception de message

```
int MPI_Sendrecv(
        cont void *     message_emis,
        int             longueur_message_emis,
        MPI_Datatype    type_message_emis,
        int             rang_dest,
        int             etiquette_message_emis,
        void *          message_recu,
        int             longueur_message_recu,
        MPI_Datatype    type_message_recu,
        int             rang_source,
        int             etiquette_message_recu,
        MPI_Comm        comm,
        MPI_Status *    statut)
```

```
int MPI_Sendrecv_replace(
        void *          message_emis_recu,
        int             longueur,
        MPI_Datatype    type,
        int             rang_dest,
        int             etiquette_message_emis,
        int             rang_source,
        int             etiquette_message_recu,
        MPI_Comm        comm,
        MPI_Status *    statut)
```

2.6 Attente de la fin d'une communication non bloquante

```
int MPI_Wait(MPI_Request * requete, MPI_Status * statut)
int MPI_Test(
        MPI_Request *   requete,
        int *           drapeau,
        MPI_Status *    statut)
```

3 Communications collectives

3.1 Diffusion générale

```
int MPI_Bcast(
        void *          messsage,
        int             longueur,
        MPI_Datatype    type,
        int             rang_source,
        MPI_Comm        comm)
```

3.2 Diffusion séléctive

```
int MPI_Scatter(
        const void *    message_a_repartir,
        int             longueur_message_emis,
        MPI_Datatype    type_message_emis,
        void *          message_recu,
        int             longueur_message_recu,
        MPI_Datatype    type_message_recu,
        int             rang_source,
        MPI_Comm        comm)
```

3.3 Collecte

```
int MPI_Gather(
        const void *    message_emis,
        int             longueur_message_emis,
        MPI_Datatype    type_message_emis,
        void *          message_collecte,
        int             longueur_message_recu,
        MPI_Datatype    type_message_recu,
        int             rang_dest,
        MPI_Comm        comm)
```

```
int MPI_Allgather(
        const void *    message_emis,
        int             longueur_message_emis,
        MPI_Datatype    type_message_emis,
        void *          message_collecte,
        int             longueur_message_recu,
        MPI_Datatype    type_message_recu,
        MPI_Comm        comm)
```

3.4 Collecte et diffusion

```
int MPI_Alltoall(
        const void *    message_a_repartir,
        int             longueur_message_emis,
        MPI_Datatype    type_message_emis,
        void *          message_collecte,
        int             longueur_message_recu,
        MPI_Datatype    type_message_recu,
        MPI_Comm        comm)
```

3.5 Réduction

```
int MPI_Reduce(
        const void *    message_emis,
        void *          message_recu,
        int             longueur,
        MPI_Datatype    type,
        MPI_Op          operation,
        int             rang_dest,
        MPI_Comm        comm)
```

operation ≡ MPI_MAX | MPI_MIN | MPI_SUM | MPI_PROD | MPI_BAND | MPI_BOR | MPI_BXOR | MPI_LAND | MPI_LOR | MPI_LXOR

MPI_Scan