**MPI - The Message Passing Interface Library**

**Introduction**

MPI is a message passing interface used for parallel processing in distributed memory systems. There are language specific bindings that allow programs written in C, C++, FORTRAN77 and FORTRAN90 to execute in parallel by calling the appropriate library routines.

A single user program is prepared, but is run on multiple processes. Each instance of the program is assigned a unique process identifier, so that it "knows" which process it is. This allows the same program to be executed, but for different things to happen in each process. Frequently, the user sets up one process as a master, and the others as workers, but this is not necessary. Each process has its own set of data, and can communicate directly with other processes to pass data around.

Because the data is distributed, it is likely that a computation on one process will require that a data value be copied from another process. Thus, if process A needs the value of data item X that is stored in the memory of process B, then the program must include lines that say something like:

```c
if ( I am processor A ) then
    call MPI_Send ( X )
else if ( I am processor B ) then
    call MPI_Recv ( X )
end
```

It should be clear that a program using MPI to execute in parallel will look much different from a corresponding sequential version. The user must divide the problem data among the different processes, rewrite the algorithm to divide up work among the processes, and add explicit calls to transfer values as needed from the process where a data item "lives" to a process that needs that value.

**Using MPI with C or C++**

A C or C++ file that uses MPI routines or constants must include the line

```c
include "mpi.h"
```

Here is a sample C program that uses MPI:

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

#include "mpi.h"

int main ( int argc, char *argv[] )
{
    int ierr;
    int master = 0;
    int my_id;
    int num_procs;
    /*
    Initialize MPI.
    */
    ierr = MPI_Init ( &argc, &argv );
    /*
    Get the number of processes.
    */
    ierr = MPI_Comm_size ( MPI_COMM_WORLD, &num_procs );
    /*
    Get the individual process ID.
    */
    ierr = MPI_Comm_rank ( MPI_COMM_WORLD, &my_id );
    /*
    Print a message.
    */
    if ( my_id == master )
    {
        printf ( "\n" );
        printf ( "HELLO_WORLD - Master process:\n" );
        printf ( " A simple C program using MPI.\n" );
        printf ( "\n" );
        printf ( " The number of processes is %d\n", num_procs );
    }

    printf ( "\n" );
    printf ( " Process %d says 'Hello, world!'\n", my_id );

    /*
    Shut down MPI.
    */
    ierr = MPI_Finalize ( );

    return 0;
}

Using MPI with Fortran

A FORTRAN77 program, subroutine or function that uses MPI must include the line

    include "mpif.h"

which defines certain parameters and function interfaces.
A FORTRAN90 program, subroutine or function may, instead of an `include` statement, access the MPI library by the line

```fortran
use mpi
```

However, this does **not** seem to be acceptable to the current IBM Fortran compilers. Thus, for now, FORTRAN90 programs on the IBM systems will need to continue to use the old style `include` statement.

Here is a sample FORTRAN90 program that uses MPI:

```fortran
program hello

    include 'mpif.h'

    integer error
    integer, parameter :: master = 0
    integer num_procs
    integer world_id

    ! Initialize MPI.
    !
    ! Get the number of processes.
    !
    ! Get the individual process ID.
    !
    ! Print a message.
    !
    if ( world_id == master ) then
        print *, ' ',
        print *, 'HElLO_WORLd - Master process:'
        print *, ' A simple FORTRAN 90 program using MPI.'
        print *, ' The number of processes is ', num_procs
    end if

    ! Shut down MPI.
    !
    call MPI_Finalize ( error )

stop
end
```
References:

- **Parallel Programming with MPI**, Peter Pacheco, Morgan Kaufman, 1996.

Source: [http://www.csit.fsu.edu/supercomputer/sp3_mpi.html](http://www.csit.fsu.edu/supercomputer/sp3_mpi.html)