

Procesos

Procesos y tareas

Proceso: Programa en ejecución (posee recursos)

Tarea: Sinónimo de proceso o parte de un proceso

También es la entidad que se asigna a un procesador.



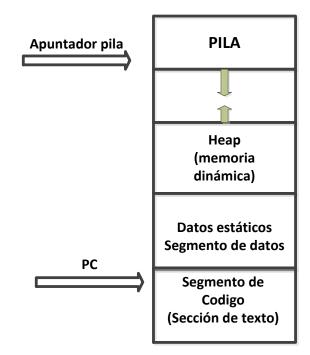
Proceso

A un proceso se le asigna un espacio de memoria y algunos otros recursos como dispositivos de entrada/salida.



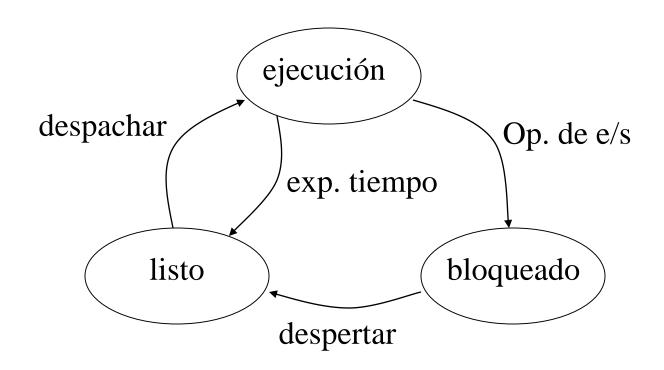
Proceso

Un proceso es una entidad formada por los siguientes elementos principales:





Estados de un proceso (ambiente concurrente)





Hilos (hebras. threads)

Un hilo no es más que la habilidad de un proceso o programa para dividirse en varios procesos ligeros de ejecución simultáneos o aparentemente simultáneos.



Un proceso, un hilo



Un proceso, varios hilos

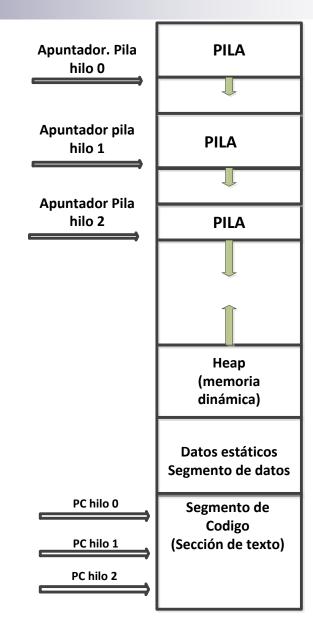


Varios procesos, un hilo



Varios procesos, varios hilos por proceso

Hilo



Introducción a MPI Message Passing Interface

Facultad Ingeniería- UNAM



MPI Message Passing Interface

- MPI es una interfaz de paso de mensajes que representa un esfuerzo prometedor de mejorar la disponibilidad de un software altamente eficiente y portable para satisfacer las necesidades actuales
- en la computación de alto rendimiento a través de la definición de un estándar de paso de mensajes universal.

William D. Gropp et al.



MPI

- MPI es un estándar de programación en paralelo mediante paso de mensajes
- Creado en 1993 como un estándar abierto por fabricantes y usuarios de sistemas paralelos.
- Incluye interfaces principalmente para FORTRAN, C, C++



Implementaciones de MPI:

- www-unix.mcs.anl.gov/mpi/mpich/
- www.lam-mpi.org/
- http://www.open-mpi.org/

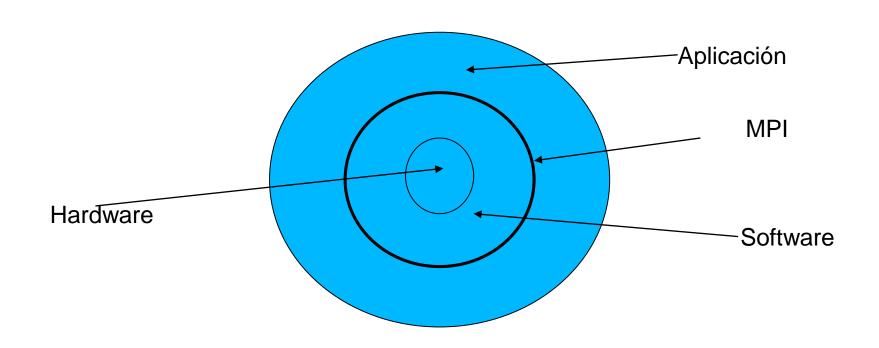
Otras direcciones:

- www-unix.mcs.anl.gov/mpi/
- www.mpi-forum.org/



- Versions: 1.0 ("94), 1.1 ("95), 1.2 ("97), 1.3 ("08)
- Basic Message Passing Concepts
- **2.0** ("97), 2.1 ("08)
- **2.2** ("09)
- 3.0 ("12) Several additions to react to new challenges
- **3.1** ("15)
- **4.0** (,,??)

Ubicación de MPI en el proceso de programación de aplicaciones paralelas





Modelos de programación

- Hardware
 - Memoria Distribuida
 - Memoria Compartida (algunas implementaciones)
 - ☐ Hibrido
 - □ PGAs

MPI no es modelo de programación

Paralelismo explicito



Funcionamiento

La unidad básica en MPI son los procesos.



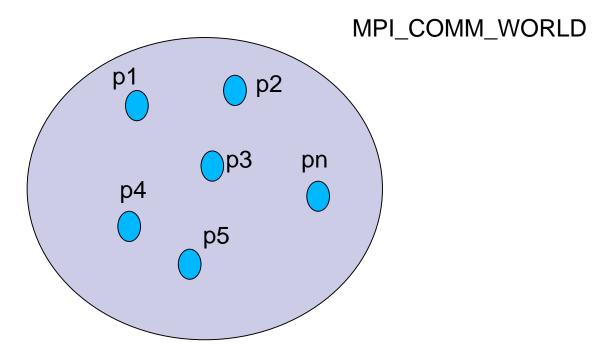
Funcionamiento

- Cada proceso se le asigna un identificador interno propio de MPI (rank).
- Tienen espacios de memoria independientes.
- Intercambio de información por paso de mensajes.
- Define varias formas de comunicación.
 Introduce en concepto de comunicadores



Comunicadores

■ ¿Qué es un comunicador?





Comunicadores

- MPI agrupa los procesos implicados en una ejecución paralela en comunicadores.
- Un comunicador agrupa a procesos que pueden intercambiarse mensajes.
- El comunicador MPI_COMM_WORLD está creado por defecto y engloba a todos los procesos.



Funciones en MPI

- C: Los nombres de todas las funciones de MPI comienzan con el prefijo MPI_. Los tipos y funciones definidas en MPI tienen una letra mayúscula después del prefijo; las letras siguientes son minúsculas.
- C++ : Todas las funciones y clases están en el espacio de nombres MPI, de tal manera que para referirnos a una función de MPI se usa MPI::Xxxxx.
- Fortran : Los nombres de todas las funciones de MPI comienzan con el prefijo MPI_ y todas las demás letras son mayúsculas.



Funciones Básicas

- MPI 1.2 tiene 129 funciones. Las funciones principales de MPI son (En C):
 - MPI_Init
 - MPI_Finalize
 - MPI_Comm_size
 - MPI_Comm_rank
 - MPI_Send
 - MPI_Recv



- mpi.h y mpif.h proveen de las definiciones básicas de MPI (funciones y tipos).
- Para iniciar MPI se usa:
 - int MPI_Init(int *argc, char **argv); (C)
 - □ void MPI::Init(int& argc, char**& argv); (C++)
 - □ INTEGER IERR y MPI_INIT(IERR) (Fortran)
- Para finalizar MPI se usa:
 - int MPI_Finalize(void); (C)
 - □ void MPI::Finalize(); (C++)
 - □ INTEGER IERR y MPI_FINALIZE(IERR) (Fortran)



Argumentos de las funciones en MPI

- E-IN: la función lo usa pero no lo actualiza.
- S-OUT: se puede actualizar este valor
- E/S-INOUT: se usa y lo actualiza

- Retorno de la función
 - ■MPI_SUCCES



MPI_Init, MPI_Finalize

- int MPI_Init(int *argc, char **argv)
 - Primera llamada de cada uno de los procesos
 MPI
 - □ Establece entorno
 - Un proceso solo puede hacer una llamada MPI_INIT
- int MPI_Finalize(void)
 - □ Termina la ejecución en MPI
 - □ Libera los recursos utilizados por MPI



MPI_Comm_rank (comm, &rank);

□ Devuelve en *rank* el identificador del proceso dentro del comunicador comm especificado

MPI_Comm_size (comm, &size);

□ Devuelve en *size* el número de procesos del comunicador especificado.

7

Estructura de un programa en MPI

MPI include file

.

Initialize MPI environment

.

Do work and make message passing calls

-

•

Terminate MPI Environment

Cómputo de Alto Desempeño - Elba Karen Sáenz García (C) v

■ Ejemplo primer programa en clase

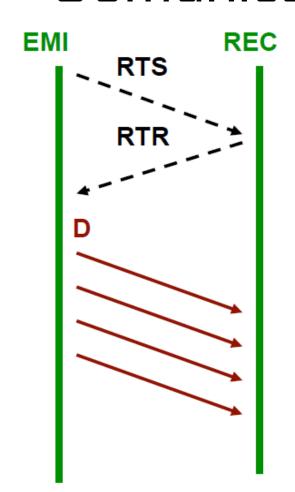


Dos modelos de Comunicación

- Una función bloqueante mantiene a un proceso bloqueado hasta que la operación solicitada finalice.
- □ Una función no bloqueante supone simplemente "encargar" al sistema la realización de una operación, recuperando el control inmediatamente. El proceso tiene que preocuparse, más adelante, de averiguar si la operación ha finalizado o no.



Comunicación



- Petición de transmisión
- Aceptación de transmisión
- Envío de datos (de buffer de usuario a buffer de usuario



Funciones Básicas de Comunicación Punto a Punto c

int MPI_Send(void *buf, int count, MPI Datatype dtype, int dest,int tag, MPI Comm comm)

int MPI_Recv(void *buf, int count, MPI Datatype dtype,
int src, int tag, MPI Comm comm,

MPI Status *status)

Argumentos

buf: Dirección donde comienza el mensaje.

count: número de elementos del mensaje.

datatype: tipo del mensaje.

dest/source: posición relativa del proceso fuente/destino dentro del comunicador. MPI_ANY_SOURCE: permite recibir mensaje de cualquier fuente.

tag: etiqueta del mensaje.

MPI_ANY_TAG: cualquier etiqueta.

comm: comunicador del proceso.



Tabla 1. Tipos de datos MPI

Tipos MPI	Tipos C equivalentes
MPI CHAR	signed char
MPI SHORT	signed short int
MPI INT	signed int
MPI LONG	signed long int
MPI UNSIGNED CHAR	unsigned char
MPI UNSIGNED SHORT	unsigned short int
MPI_UNSIGNED	unsigned int
MPI UNSIGNED LONG	unsigned long int
MPI FLOAT	float
MPI_DOUBLE	double
MPI_LONG_DOUBLE	long double
MPI_BYTE	Sin equivalente

10

Ejemplo de comunicación punto a punto

м

Ejemplo 3

