

C  puto Paralelo

Manual de Instalaci  n de MPI

Valeria Jahzeel Casta  n H  rn  ndez

29 de abril de 2024

1. Instalaci  n y configuraci  n

Si estas en Linux, abre una terminal y ejecuta el comando.

```
sudo apt-get update
```

El comando sirve para descargar la informaci  n m  s reciente sobre los paquetes de software disponibles, incluyendo sus versiones, dependencias y cambios.

Posteriormente necesitamos descargar el **compilador GNU de fortran** mediante el comando

```
sudo apt-get install gfortran
```

Hecho esto, creamos un directorio llamado mpich3

```
mkdir home/user/mpich3
```

Nota: user corresponde al nombre de usuario que tienes registrado en tu sistema

Procedemos a entrar a la carpeta

```
cd mpich3
```

Y a todo esto...   qu   es MPICH3?

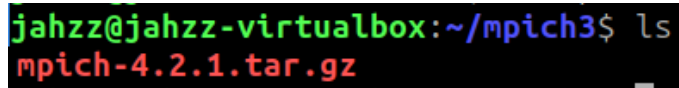
MPICH3 es una implementaci  n de alto rendimiento del est  ndar MPI, que permite a los programas comunicarse entre s   a trav  s de m  ltiples procesadores u ordenadores para tareas grandes. Esto posibilita la escritura de programas paralelos que aprovechan la potencia de m  ltiples m  quinas.

Ahora que ya lo sabemos, seguimos con la instalaci  n de la versi  n mas reciente de MPICH3,

en este caso es la versi  n 4.2.1

```
wget http://www.mpich.org/static/downloads/4.2.1/mpich-4.2.1.tar.gz
```

Esperamos a que se instale (es normal que tarde mucho) y una vez instalada, verificamos que est   en la carpeta utilizando el comando `ls`.



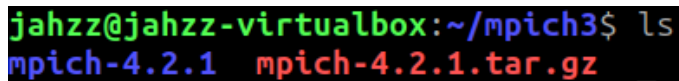
```
jahzz@jahzz-virtualbox:~/mpich3$ ls  
mpich-4.2.1.tar.gz
```

Figura 1: Verificaci  n de la instalaci  n de mpich

Luego ejecutamos el comando

```
tar xzf mpich-4.2.1.tar.gz
```

para descomprimir el archivo, si nuevamente vemos el contenido de la carpeta (`ls`) veremos algo as  



```
jahzz@jahzz-virtualbox:~/mpich3$ ls  
mpich-4.2.1  mpich-4.2.1.tar.gz
```

Figura 2: Carpeta mpich3 despu  s de descomprimir los archivos

Ahora creamos tres nuevos directorios

```
sudo mkdir /home/rpimpi/  
sudo mkdir /home/rpimpi/mpich3-install  
sudo mkdir /home/username/mpich_build
```

Una vez hecho esto, nos movemos a la carpeta **mpich_build**

```
cd /home/username/mpich_build
```

Ejecutamos el comando

```
sudo /home/username/mpich3/mpich-4.2.1/configure --prefix=/home/rpimpi/mpich3-install
```

para configurar y preparar la instalaci  n de MPICH3. Esperamos a que se instalen todos los archivos.

NOTA: si da un error de bootstrapping o algo similar ejecuta el comando

```
sudo apt-get install build-essential
```

Una vez instalado todo, ejecuta el comando

```
sudo make
```

Al finalizar, te debe dar algo similar a esto

```
Making install in examples
make[2]: Entering directory `/home/pi/mpich_build/examples'
make[3]: Entering directory `/home/pi/mpich_build/examples'
make[3]: Nothing to be done for `install-exec-am'.
make[3]: Nothing to be done for `install-data-am'.
make[3]: Leaving directory `/home/pi/mpich_build/examples'
make[2]: Leaving directory `/home/pi/mpich_build/examples'
make[1]: Leaving directory `/home/pi/mpich_build'
```

Figura 3: Desp  es de ejecutar sudo make

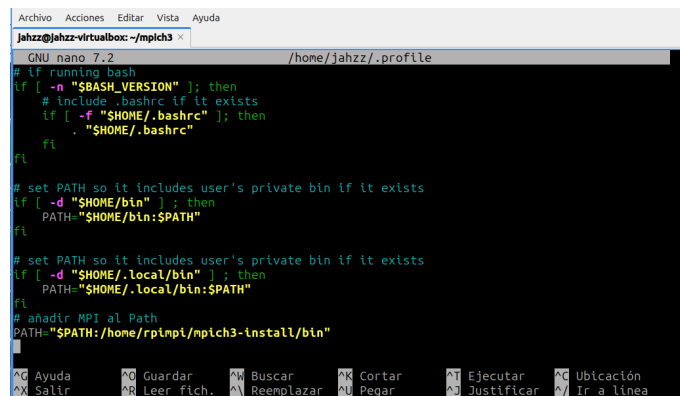
Si no te da algo similar entonces ejecuta el comando

```
sudo make install
```

Una vez que se terminan de instalar los archivos se ejecuta

```
sudo nano /home/username/.profile
```

Te va a mostrar una pantalla similar a esta



```
GNU nano 7.2 /home/jahzz/.profile
# If running bash
if [ -n "$BASH_VERSION" ]; then
# include .bashrc if it exists
if [ -f "$HOME/.bashrc" ]; then
. "$HOME/.bashrc"
fi
fi

# set PATH so it includes user's private bin if it exists
if [ -d "$HOME/bin" ] ; then
PATH="$HOME/bin:$PATH"
fi

# set PATH so it includes user's private bin if it exists
if [ -d "$HOME/.local/bin" ] ; then
PATH="$HOME/.local/bin:$PATH"
fi

# a  adir MPI al Path
PATH="$PATH:/home/rpimpi/mpich3-install/bin"
```

Figura 4: Pantalla del PATH

Vamos a a  adir el MPI al PATH escribiendo lo siguiente

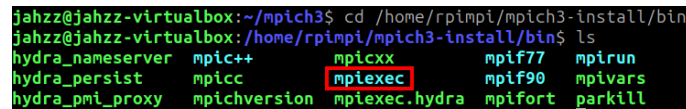
```
PATH = "$PATH:/home/rpimpi/mpich3-install/bin"
```

Hacemos **ctrl + X** para salir y guardamos los cambios (S)

Ahora nos movemos a la ruta del PATH

```
cd /home/rpimpi/mpich3-install/bin
```

Y verificamos que archivos hay en la carpeta (**ls**), el archivo que debemos buscar debe ser **mpiexec**



```
jahzz@jahzz-virtualbox:~/mpich3$ cd /home/rpimpi/mpich3-install/bin
jahzz@jahzz-virtualbox:/home/rpimpi/mpich3-install/bin$ ls
hydra_nameserver  mpicxx      mpif77  mpiexec  mpiexec.hydra  mpifort  parkill
hydra_persist     mpicc       mpif90  mpiexec
hydra_pmi_proxy   mpichversion
```

Figura 5: Verificar que **mpiexec** est   en la carpeta del PATH

Ahora regresamos a la carpeta ra  z

```
cd ~
```

Y ejecutamos

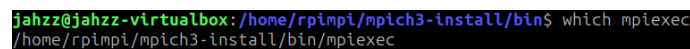
```
source ~/.profile
```

para recargar el perfil y se apliquen los cambios que hemos hecho.

Luego colocamos el comando

```
which mpiexec
```

para ver si se encontr   el archivo en la ruta del PATH que especificamos. Debe salir algo as  :



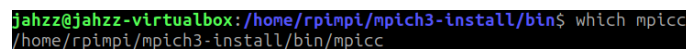
```
jahzz@jahzz-virtualbox:/home/rpimpi/mpich3-install/bin$ which mpiexec
/home/rpimpi/mpich3-install/bin/mpiexec
```

Figura 6: Resultado de ejecutar **which mpiexec**

Lo mismo hacemos con

```
which mpicc
```

D  ndonos un resultado similar:



```
jahzz@jahzz-virtualbox:/home/rpimpi/mpich3-install/bin$ which mpicc
/home/rpimpi/mpich3-install/bin/mpicc
```

Figura 7: Resultado de ejecutar **which mpic**

2. Pruebas

Regresamos a la carpeta raiz

```
cd /home/username
```

Y creamos una carpeta para las pruebas

```
mkdir mpi_testing
```

Entramos a la carpeta

```
cd mpi_testing
```

Necesitamos conocer nuestra direcci  n IP, por lo que ejecutamos alguno de estos comandos

```
ipconfig  
ip addr  
ip a
```

La IP es la que empieza por 192.XXX.X.XX en nuestro caso la IP es **192.168.1.70**

Ahora se ejecuta

```
sudo nano machinefile
```

Y va a aparecer una pantalla negra con instrucciones similares a la del PATH, en la pantalla colocamos la IP

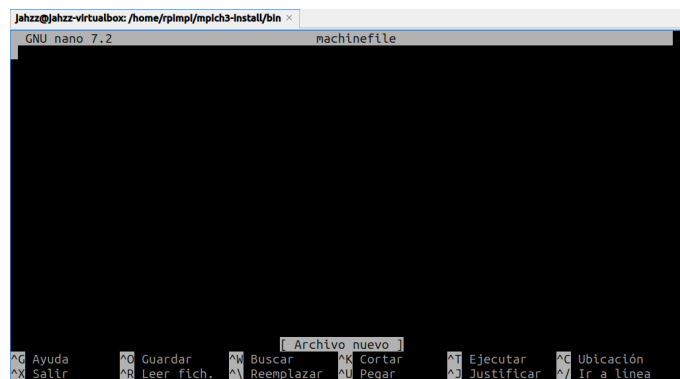


Figura 8: Pantalla en la que se coloca la IP

Presionamos **ctrl + X** para salir y guardar cambios

Si ejecutmos **ls** debe aparecer **machinefile**

```
jahzz@jahzz-virtualbox:~/mpi_testing$ ls  
machinefile
```

Figura 9: Despu  s de a  adir la IP



Figura 10: Confirmaci  n del guardado de la IP

Y si abrimos nuevamente `sudo nano machinefile` debemos ver que ah   sigue nuestra direcci  n IP

2.1. Verificar que MPI funcione correctamente

Ahora, se ejecuta

```
mpiexec -f machinefile -n 1 hostname
```

lo que nos debe dar algo similar a esto

```
jahzz@jahzz-virtualbox:~/mpi_testing$ mpiexec -f machinefile -n 1 hostname  
jahzz-virtualbox
```

Figura 11: Resultado de la prueba 1

Aqu   una peque  a explicaci  n de los comandos:

- **mpiexec:** Programa para lanzar aplicaciones MPI, inicia el entorno de ejecuci  n MPI y gestiona la comunicaci  n entre procesos.
- **-f machinefile:** Opci  n que especifica la ubicaci  n de un archivo llamado machinefile, que contiene una lista de nombres de host o direcciones IP. En este caso particular con `-n 1`, no se est   lanzando nada en esas m  quinas.
- **-n 1:** Indica a mpiexec que lance solo una instancia del programa, ejecut  ndolo en una   nica m  quina (la m  quina actual).

- **hostname:** El programa que se est   ejecutando, imprime el nombre de host de la m  quina en la que se est   ejecutando.

En resumen, lo que hace es que nos da el nombre del host del usuario 1 (que somos nosotros)

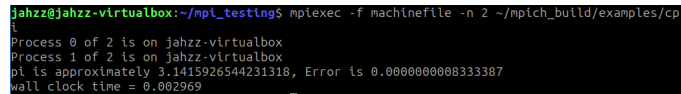
Ahora vamos a la carpeta

```
cd ~/mpi_testing
```

y ejecutamos

```
mpiexec -f machinefile -n 2 ~/mpich_build/examples/cpi
```

lo que se obtiene es una aproximaci  n del numero PI



```
jahzz@jahzz-virtualbox:~/mpi_testing$ mpiexec -f machinefile -n 2 ~/mpich_build/examples/cpi
Process 0 of 2 is on jahzz-virtualbox
Process 1 of 2 is on jahzz-virtualbox
pi is approximately 3.1415926544231318, Error is 0.0000000008333387
wall clock time = 0.002969
```

Figura 12: Resultado de la prueba 2

Aqu   la explicaci  n de los comandos usados

- **mpiexec:** Inicia el entorno de ejecuci  n MPI y gestiona la comunicaci  n entre procesos.
- **-f machinefile:** Especifica la ubicaci  n de un archivo llamado machinefile, que contiene una lista de nombres de host o direcciones IP. Estas m  quinas se utilizar  n para ejecutar el programa cpi en paralelo.
- **-n 2:** Indica a mpiexec que lance dos instancias del programa cpi. Esto crea dos procesos MPI que cooperar  n para calcular pi.
- **/mpich_build/examples/cpi:** Especifica la ubicaci  n del programa que deseas ejecutar en paralelo.

Para finalizar, ejecutamos

```
sudo shutdown -h now
```

para terminar el proceso.