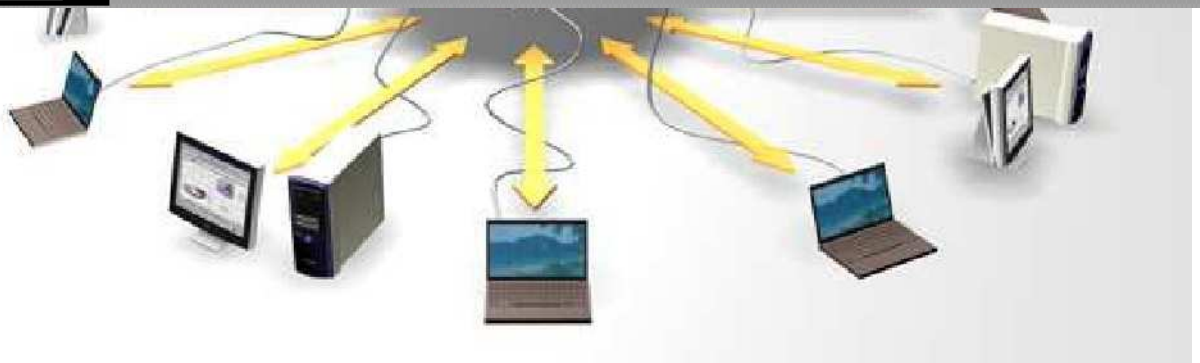


Veracruz, México Enero 2009



Instalación y configuración de MPI y openMP para los lenguajes C/C++ y Fortran usando KDevelop



Contenido

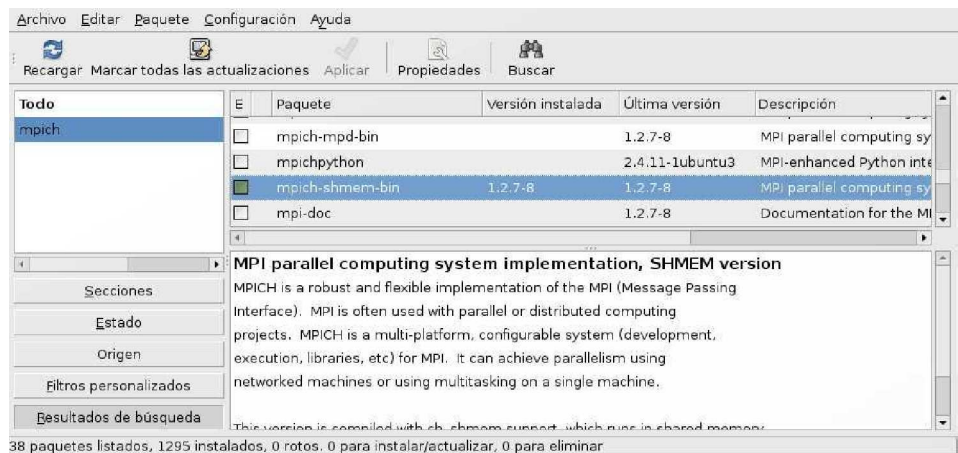
1. Configurando MPI para C/C++	3
1.1 Paquetes a instalar	3
1.2 Creando un programa en Kdevelop.....	3
2. Configurando openMP para C/C++	6
2.1 Paquetes a instalar	6
2.3 Compilando y ejecutando desde la linea de comandos	10
3. Creando el profile para C/C++	12
4. Configurando MPI para Fortran	14
4.1 Paquetes a instalar	14
4.2 Creando un programa en Kdevelop.....	14
5. Configurando openMP para Fortran.....	17
5.1 Paquetes a instalar	17
5.2 Creando un programa en Kdevelop.....	17
5.3 Compilando y ejecutando desde la línea de comandos	20
6. Creando el profile para Fortran.....	21

1. Configurando MPI para C/C++

1.1 Paquetes a instalar

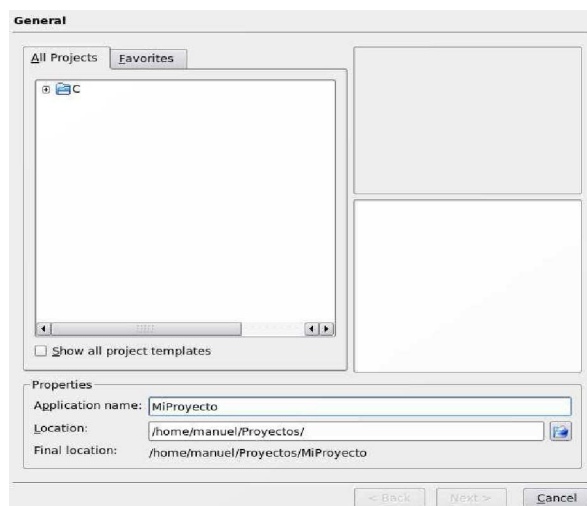
Puede usar el gestor de paquetes Synaptic para instalar los siguientes paquetes:

- binutils
- automake
- autoconf
- g++
- gcc
- m4
- libtool
- konsole
- kdevelop
- mpich-shmem-bin
- libmpich-shmem1.0-dev
- libmpich-shmem1.0gf



1.2 Creando un programa en Kdevelop

Menu Project -> New Project. Elegir proyecto de C/C++ y escribir el nombre del proyecto a crear y elegir la ubicación del mismo.

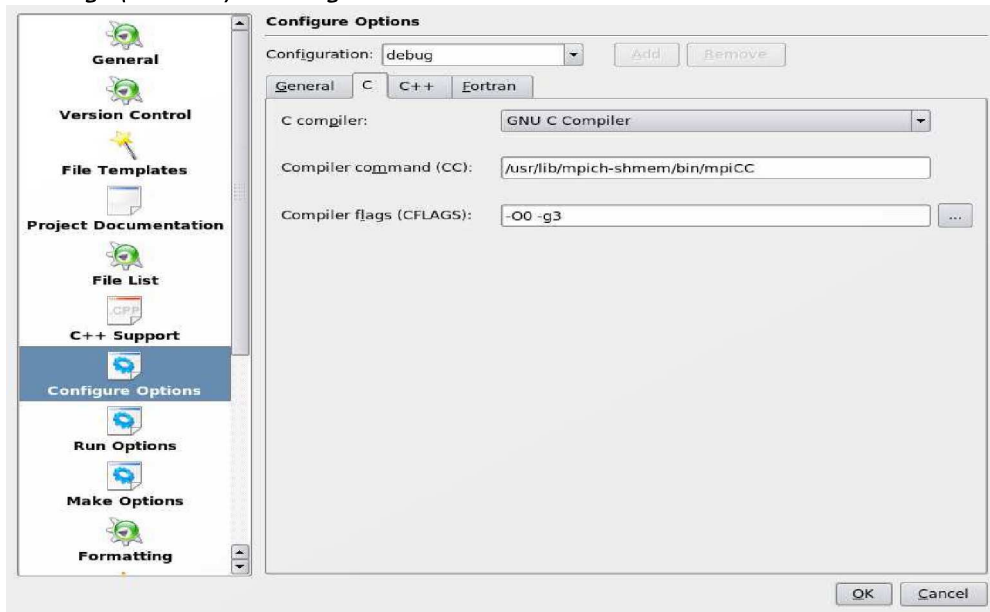


1. Configurando MPI para C/C++

Tec. Manuel Antonio Ortiz Montalvo

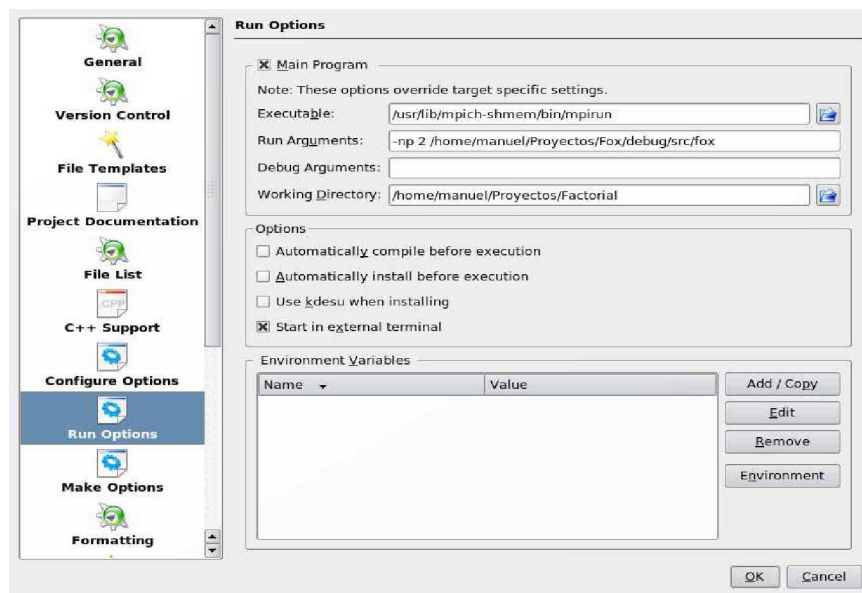
En el menú *Project -> Project Options -> Configure options*, en la pestaña C, debe configurar las siguientes opciones:

Compiler: *GNU C Compiler*
Compiler Command (CC): */usr/lib/mpich-shmem/bin/mpicc*
Compiler Flags (CFLAGS): *-O0 -g3*



En el menú *Run Options*, debe escribir la siguiente información:

Executable: */usr/lib/mpich-shmem/bin/mpirun*
Run Arguments: *-np noDenodos rutaDondeSeGeneraElEjecutableDeTuProyecto*
Working Directory: *rutaDeTuProyecto*



Posteriormente aceptar los cambios dando click en OK.

1. Configurando MPI para C/C++

Tec. Manuel Antonio Ortiz Montalvo

Escribir tu programa, pero no olvides incluir la libreria mpi.h. A continuación se muestra un código de ejemplo.

```
factorial.c
#ifdef HAVE_CONFIG_H
#include <config.h>
#endif
#include "/usr/lib/mpich-shmem/include/mpi.h"
#include <stdio.h>

int main(int argc, char *argv[])
{
    int nodos, rango;

    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MPI_COMM_WORLD, &rango);
    MPI_Comm_size(MPI_COMM_WORLD, &nodos);

    printf("Hola mundo desde el nodo %d\n", rango);

    MPI_Finalize();

    return 0;
}
```

Antes de compilar por primera vez, debes dar click en las siguientes opciones:

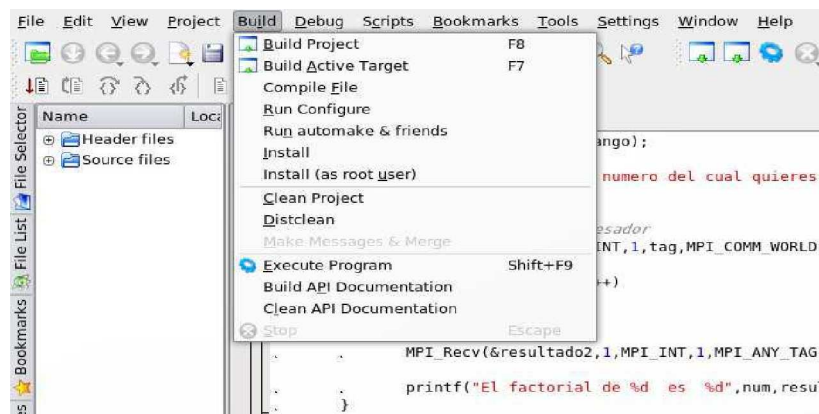
Menú Build -> Install

Menú Build -> Run Automake & Friends

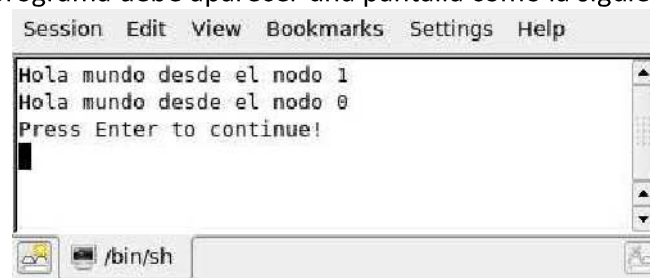
Menú Build -> Run Configure

Para compilar se dá click en Menu *Build* -> *Build Project* y posteriormente en Menu *Build* -> *Build Active Target*.

Para ejecutar tu programa click en Menu *Build* -> *Execute Program*



Después de ejecutar tu programa debe aparecer una pantalla como la siguiente:

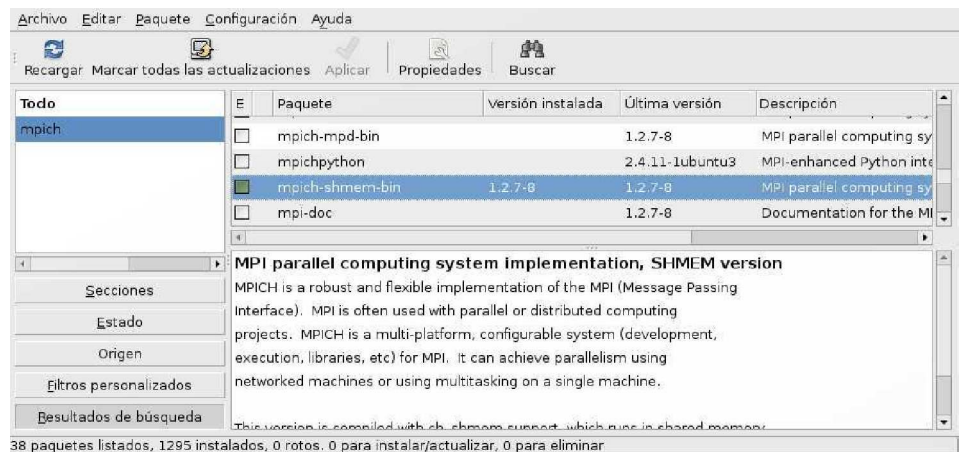


2. Configurando openMP para C/C++

2.1 Paquetes a instalar

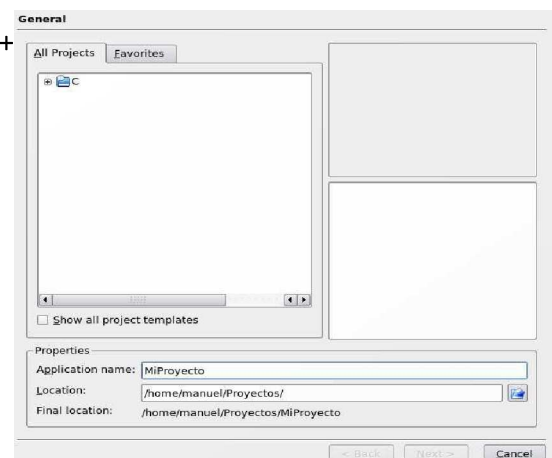
Debe instalar los siguientes paquetes, sólo si no instaló los necesarios para usar MPI.

- binutils
- automake
- autoconf
- g++
- gcc
- m4
- libtool
- konsole
- kdevelop

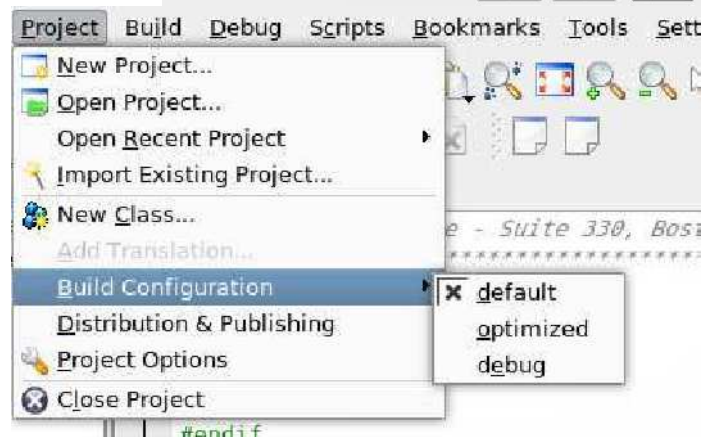


2.2 Creando un programa en Kdevelop

Menu Project -> New Project. Elegir tipo de proyecto C/C++ y escribir el nombre del proyecto a crear y elegir la ubicación del mismo.



Elegir la opción **default** del menú Project-> Build Configuration

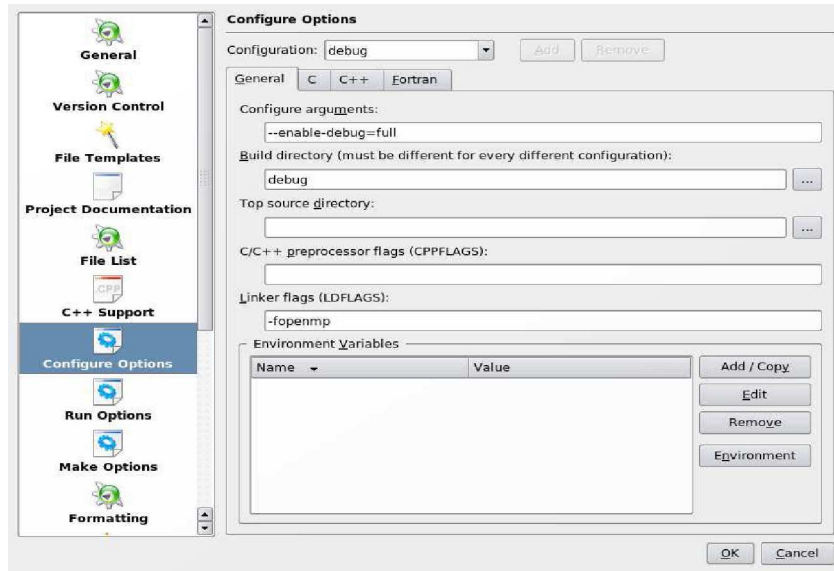


2. Configurando openMP para C/C++

Tec. Manuel Antonio Ortiz Montalvo

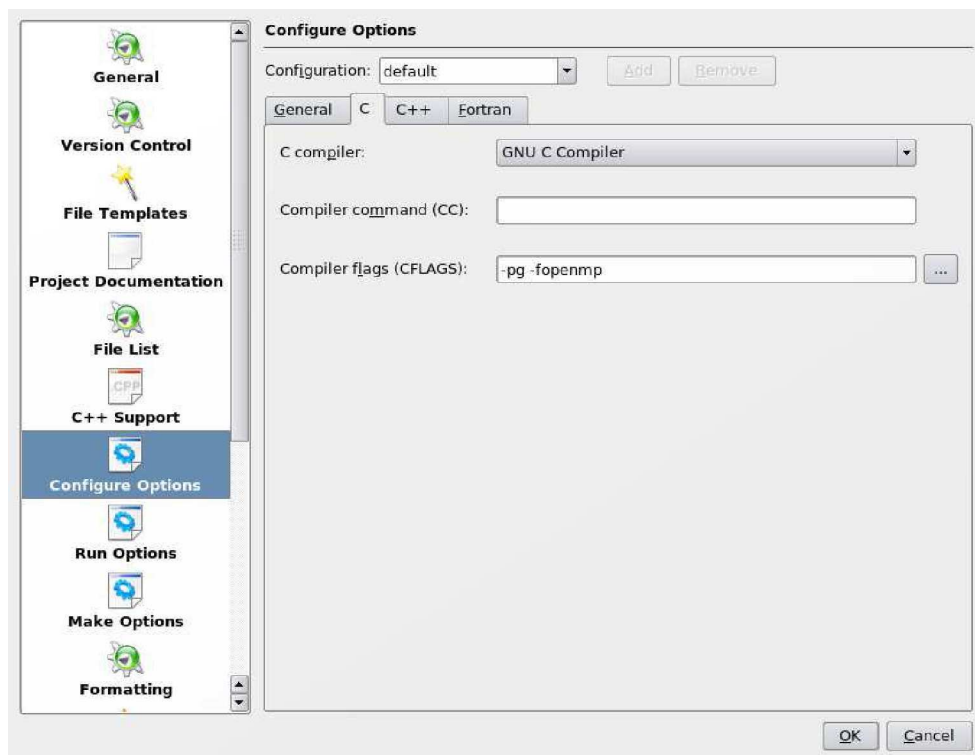
Para configurar openMP, debe dar click en Menú Project -> Project Options -> Configure Options, y en la pestaña general debe configurar la siguiente opción:

Linker Flags (LDFLAGS): -fopenmp



Y en la pestaña C, debe configurar la siguiente opción

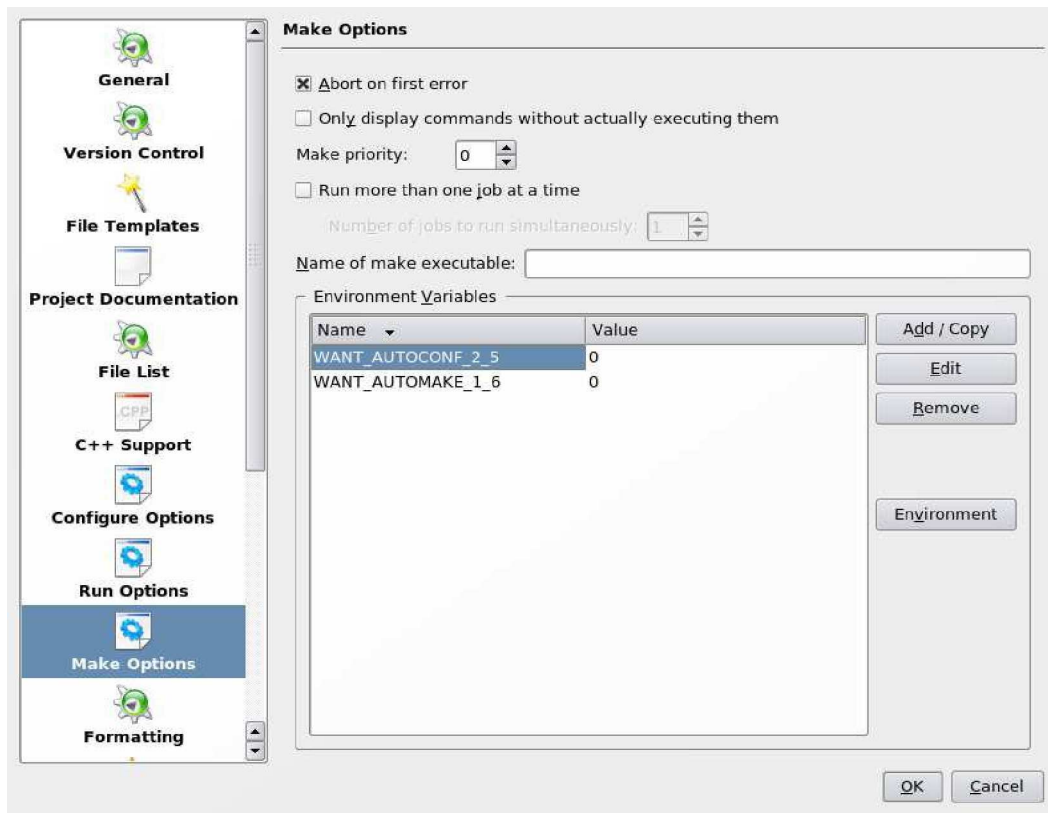
C Compiler: GNU C Compiler
Compiler Flags (CFLAGS): -pg -fopenmp



2. Configurando openMP para C/C++

Tec. Manuel Antonio Ortiz Montalvo

Elegir el menú Make Options y establecer a las variables WANT_AUTOCONF_2_5 y WANT_AUTOMAKE_1_6 los valores de 0.



Posteriormente aceptar los cambios dando click en OK.

Escribir el siguiente código de ejemplo, para demostrar como funciona openMP. Nótese que debe incluir la libreria omp.h, como se muestra en la siguiente figura.

```
pruebaomp.c
#ifdef HAVE_CONFIG_H
#include <config.h>
#endif

#include <stdio.h>
#include <stdlib.h>
#include <omp.h>

int main(int argc, char *argv[])
{
    int noHilo, totalHilos;

    omp_set_num_threads(4);
    #pragma omp parallel private(noHilo) shared(totalHilos)
    {
        totalHilos=omp_get_num_threads();
        noHilo=omp_get_thread_num();
        printf("Thread= %d de %d\n",noHilo,totalHilos);
    }
    #pragma omp end parallel

    return EXIT_SUCCESS;
}
```


2. Configurando openMP para C/C++

Tec. Manuel Antonio Ortiz Montalvo

Antes de compilar por primera vez, debes dar click en las siguientes opciones:

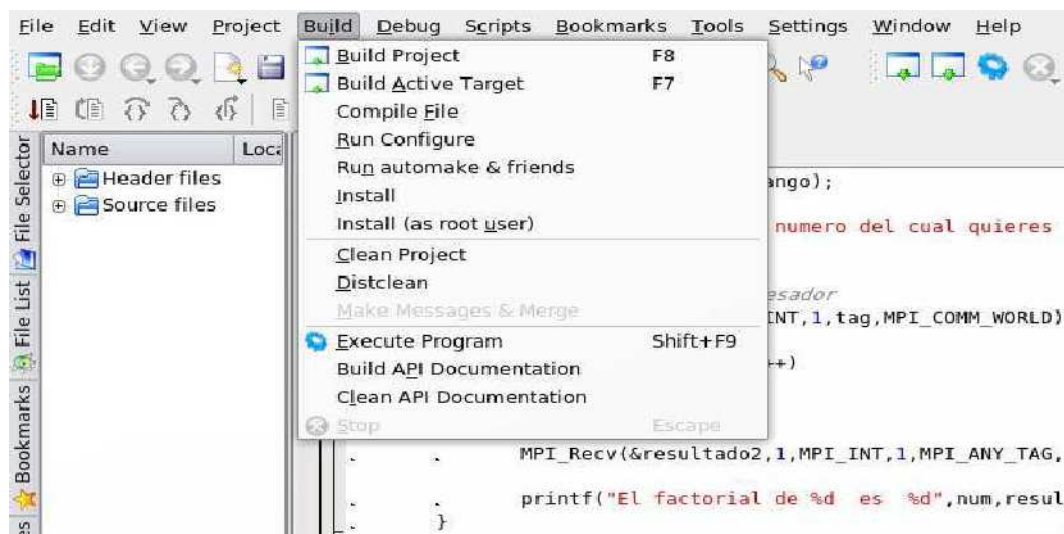
Menú Build -> Install

Menú Build -> Run Automake & Friends

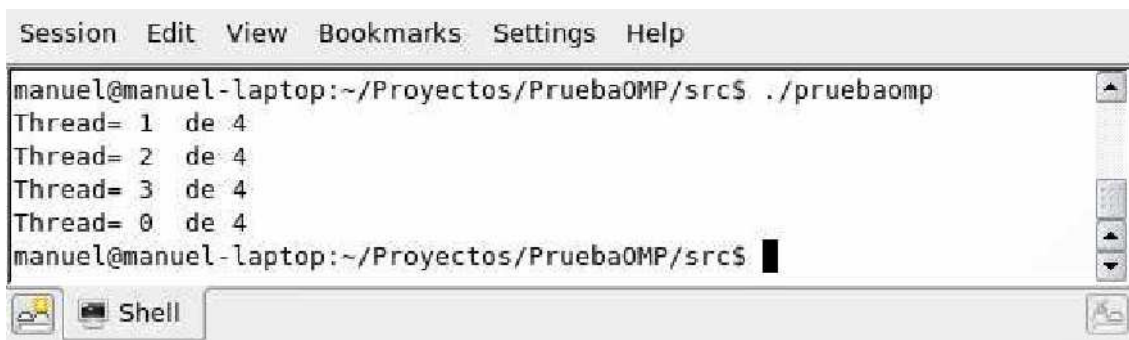
Menú Build -> Run Configure

Para compilar se dá click en *Menu Build -> Build Project* y posteriormente en *Menu Build -> Build Active Target*.

Para ejecutar tu programa click en *Menu Build -> Execute Program*

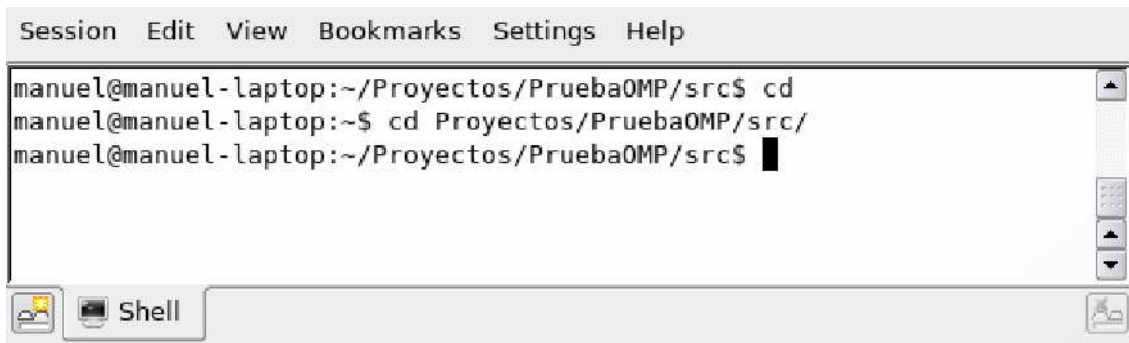


Después de ejecutar tu programa debe aparecer una pantalla como la siguiente:



2.3 Compilando y ejecutando desde la línea de comandos

Para compilar desde la línea de comandos, primero debes ubicarte en la carpeta que contiene el código fuente.



```
Session Edit View Bookmarks Settings Help
manuel@manuel-laptop:~/Proyectos/PruebaOMP/src$ cd
manuel@manuel-laptop:~$ cd Proyectos/PruebaOMP/src/
manuel@manuel-laptop:~/Proyectos/PruebaOMP/src$
```

Posteriormente ejecutar el siguiente comando para realizar la compilación

```
gcc -pg -fopenmp -o programaEjecutable codigoFuente.c
```

Dónde:

<i>-pg</i>	<i>Activa la opción de generar el Profile</i>
<i>-fopenmp</i>	<i>Activa la opción de crear ambiente paralelo</i>
<i>-o</i>	<i>Genera el archivo objeto</i>
<i>programaEjecutable</i>	<i>Nombre del archivo ejecutable, que generará la compilación</i>
<i>codigoFuente.c</i>	<i>Nombre de tu archivo fuente</i>

A continuación se muestra un ejemplo:

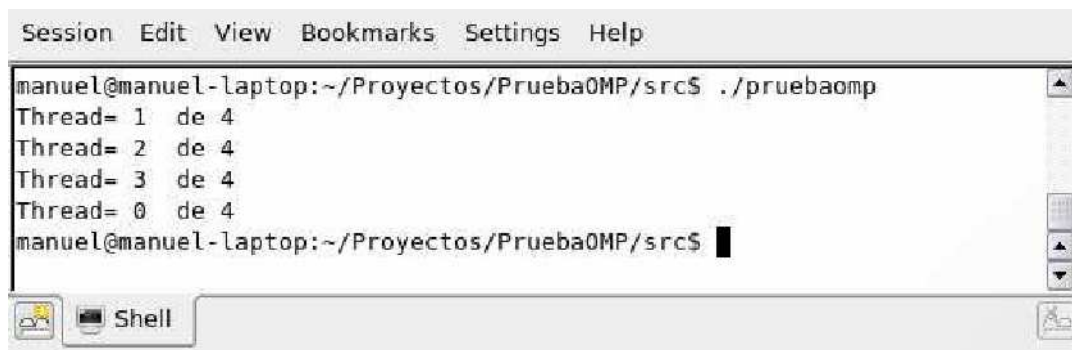


```
Session Edit View Bookmarks Settings Help
manuel@manuel-laptop:~/Proyectos/PruebaOMP/src$ gcc -pg -fopenmp -o pruebaomp pruebaomp.c
manuel@manuel-laptop:~/Proyectos/PruebaOMP/src$
```

Y ya para ejecutar tu programa, sólo debes escribir el nombre de tu ejecutable que se acaba de generar, antecedido por los signos “./”, y tu programa debe empezar a ejecutarse, como muestra el siguiente ejemplo:

2. Configurando openMP para C/C++

Tec. Manuel Antonio Ortiz Montalvo



```
Session Edit View Bookmarks Settings Help
manuel@manuel-laptop:~/Proyectos/PruebaOMP/src$ ./pruebaomp
Thread= 1 de 4
Thread= 2 de 4
Thread= 3 de 4
Thread= 0 de 4
manuel@manuel-laptop:~/Proyectos/PruebaOMP/src$
```

3. Creando el profile para C/C++

Si se desea hacer un análisis de la ejecución del código, los tiempos que demoró la ejecución de cada método, y las veces que se llamo cada uno, entre mucha otra información, se puede crear el profile.

Para crearlo, primeramente el programa se tiene que compilar activando la bandera pg:

```
gcc -pg -openmp -o programa archivoFuente.f
```



A screenshot of a terminal window with a menu bar (Session, Edit, View, Bookmarks, Settings, Help) and a title bar (Shell). The terminal shows the command `gcc -pg -fopenmp -o prueba ompconc.c` being executed in a directory `~/Proyectos/ompConC/src`. The prompt is `manuel@manuel-laptop:~/Proyectos/ompConC/src$`.

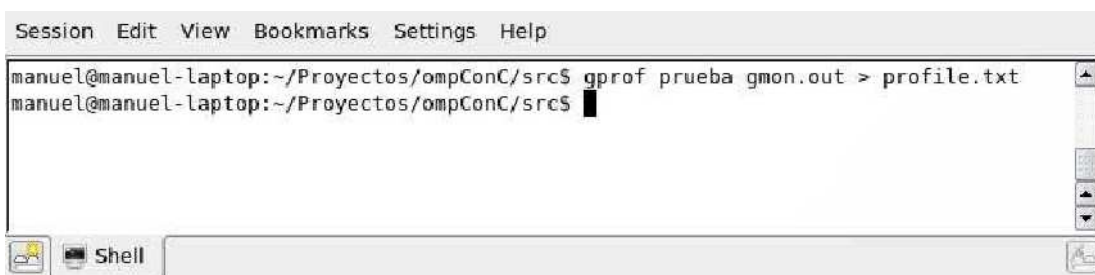
Si compiló correctamente, se tuvo que generar un archivo llamado gmon.out en la carpeta de tu código fuente.

Posteriormente, se crea tu archivo profile, de la siguiente manera:

```
gprof programa gmon.out > archivo.txt
```

Donde:

programa: Es el nombre del ejecutable que se generó durante la compilación
gmon.out Archivo que se generó durante la compilación con la bandera pg
archivo.txt Nombre del archivo que generará con la información del profile



A screenshot of a terminal window with a menu bar (Session, Edit, View, Bookmarks, Settings, Help) and a title bar (Shell). The terminal shows the command `gprof prueba gmon.out > profile.txt` being executed in a directory `~/Proyectos/ompConC/src`. The prompt is `manuel@manuel-laptop:~/Proyectos/ompConC/src$`.

Para visualizar el archivo que generó, sólo tienes que escribir el comando:

```
edit nombreArchivo
```

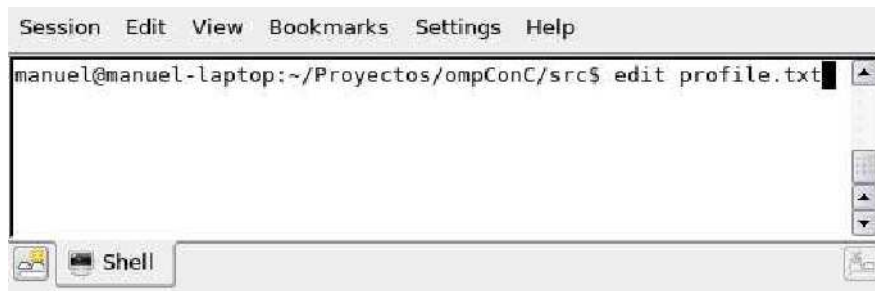
Donde:

nombreArchivo: Es el nombre del archivo a visualizar, que es el que generamos con el comando `gprof`

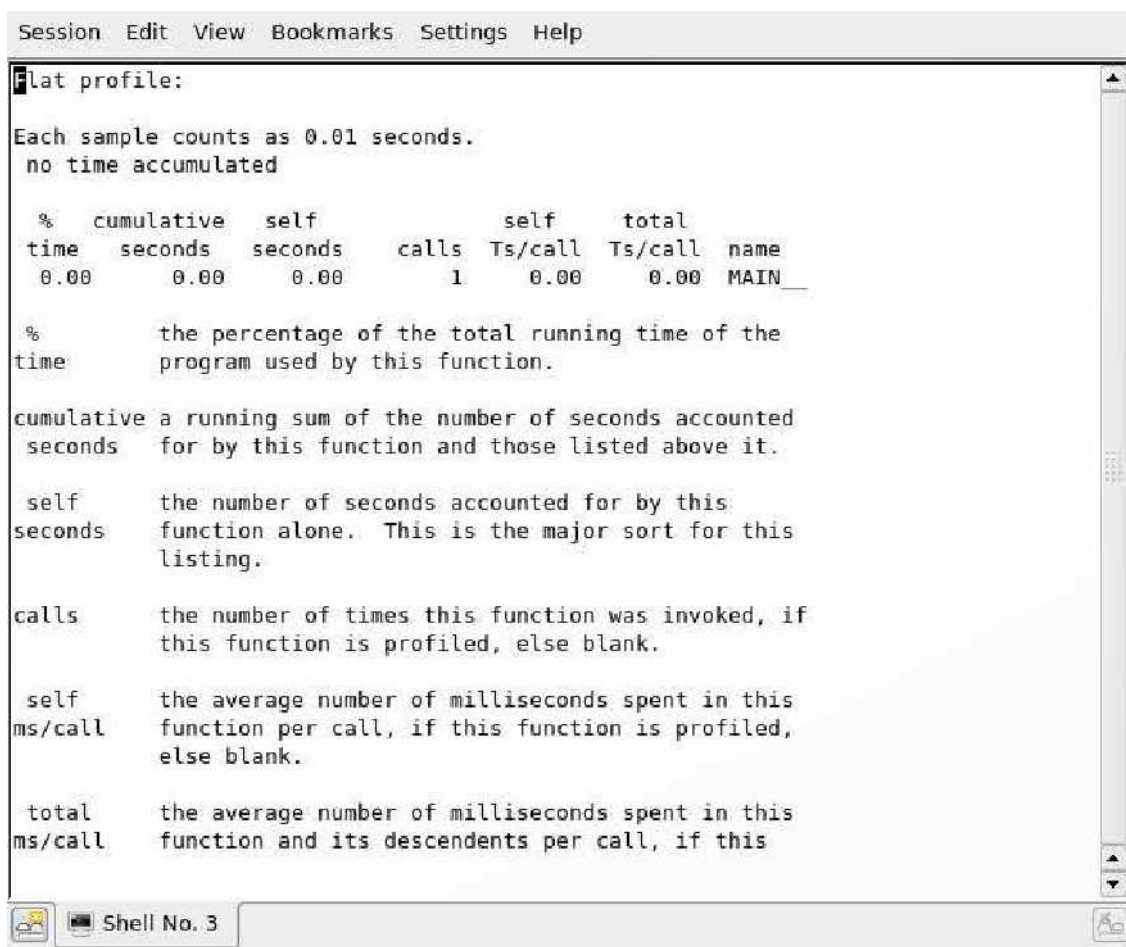
3. Creando el profile para C/C++

Tec. Manuel Antonio Ortiz Montalvo

A continuación se muestra un ejemplo de la ejecución del comando, y el resultado del mismo.



```
Session Edit View Bookmarks Settings Help
manuel@manuel-laptop:~/Proyectos/ompConC/src$ edit profile.txt
```



```
Session Edit View Bookmarks Settings Help
Flat profile:
Each sample counts as 0.01 seconds.
no time accumulated

% cumulative self      self   total
time  seconds seconds  calls Ts/call Ts/call  name
0.00    0.00    0.00      1    0.00    0.00  MAIN_

%           the percentage of the total running time of the
time        program used by this function.

cumulative  a running sum of the number of seconds accounted
seconds     for by this function and those listed above it.

self        the number of seconds accounted for by this
seconds     function alone.  This is the major sort for this
           listing.

calls       the number of times this function was invoked, if
           this function is profiled, else blank.

self        the average number of milliseconds spent in this
ms/call     function per call, if this function is profiled,
           else blank.

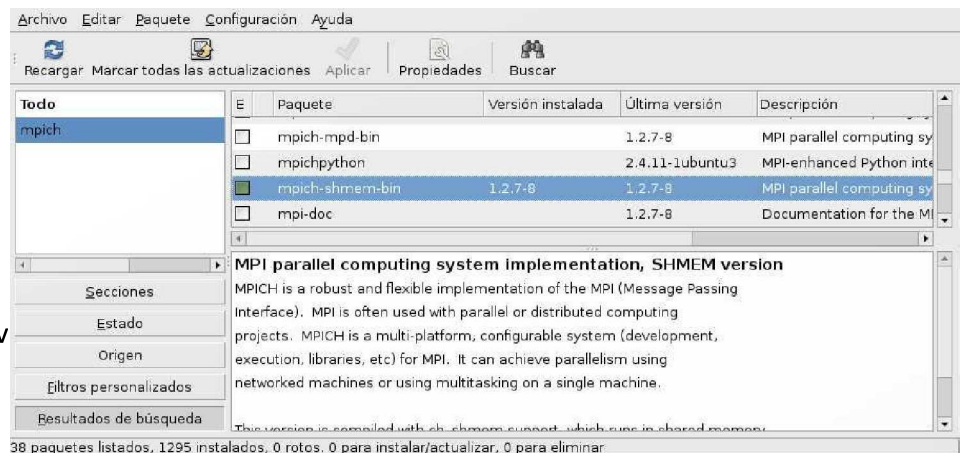
total       the average number of milliseconds spent in this
ms/call     function and its descendents per call, if this
```

4. Configurando MPI para Fortran

4.1 Paquetes a instalar

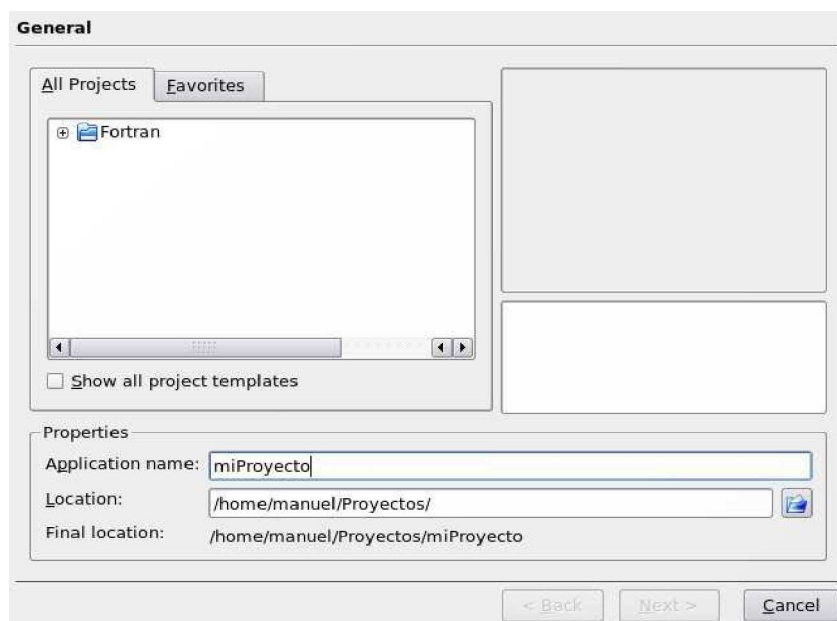
Puede usar el gestor de paquetes Synaptic para instalar los siguientes paquetes:

- binutils
- automake
- autoconf
- gfortran
- m4
- libtool
- konsole
- kdevelop
- mpich-shmem-bin
- libmpich-shmem1.0-dev
- libmpich-shmem1.0gf



4.2 Creando un programa en Kdevelop

Menu Project -> New Project. Elegir proyecto de Fortran y escribir el nombre del proyecto a crear y elegir la ubicación del mismo.

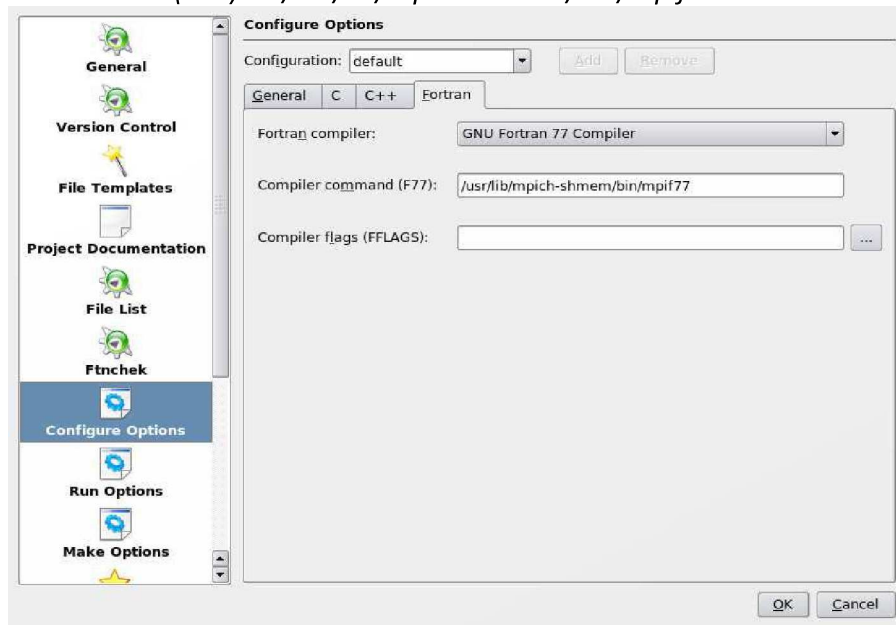


4. Configurando MPI para Fortran

Tec. Manuel Antonio Ortiz Montalvo

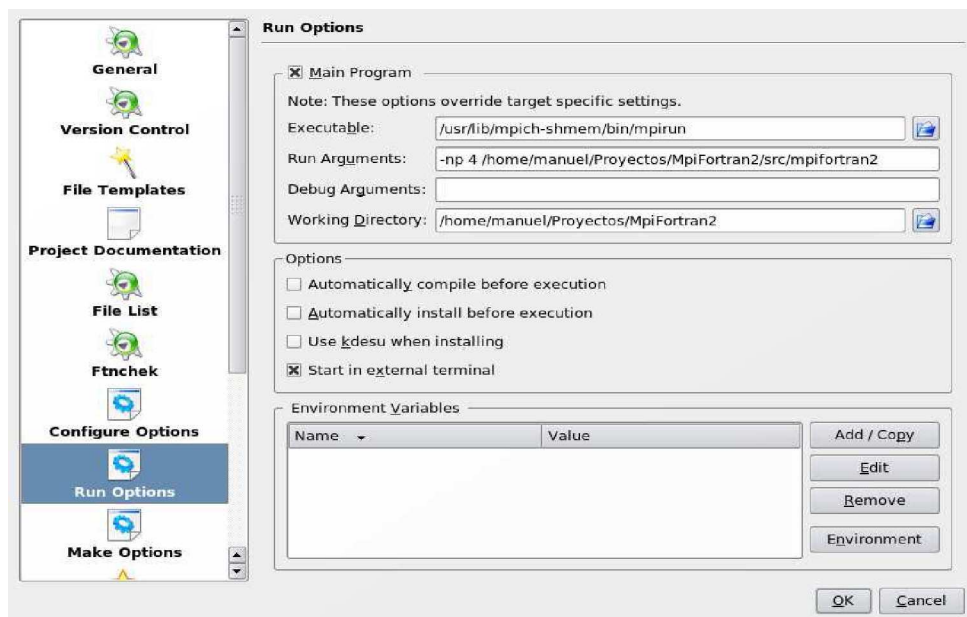
En el menú *Project -> Project Options -> Configure options*, en la pestaña Fortran, debe configurar las siguientes opciones:

Compiler: GNU Fortran Compiler
Compiler Command (F77): /usr/lib/mpich-shmem/bin/mpif77



En el menú *Run Options*, debe escribir la siguiente información:

Executable: /usr/lib/mpich-shmem/bin/mpirun
Run Arguments: -np 4 /home/manuel/Proyectos/MpiFortran2/src/mpifortran2
Working Directory: rutaDeTuProyecto



Posteriormente aceptar los cambios dando click en OK.

4. Configurando MPI para Fortran

Tec. Manuel Antonio Ortiz Montalvo

Escribir tu programa y no olvides incluir la libreria mpif.h, como se muestra en la siguiente figura



```
mpifortran2.f
c
c Ejemplo del Hola mundo
c
program hello
include "/usr/lib/mpich-shmem/include/mpif.h"
integer tamaño,rango,error
call MPI_Init(error)
call MPI_Comm_Rank(MPI_COMM_WORLD,rango,error)
print *, "Hola mundo desde ",rango
call MPI_Finalize(error)
stop
end
```

Antes de compilar por primera vez, debes dar click en las siguientes opciones:

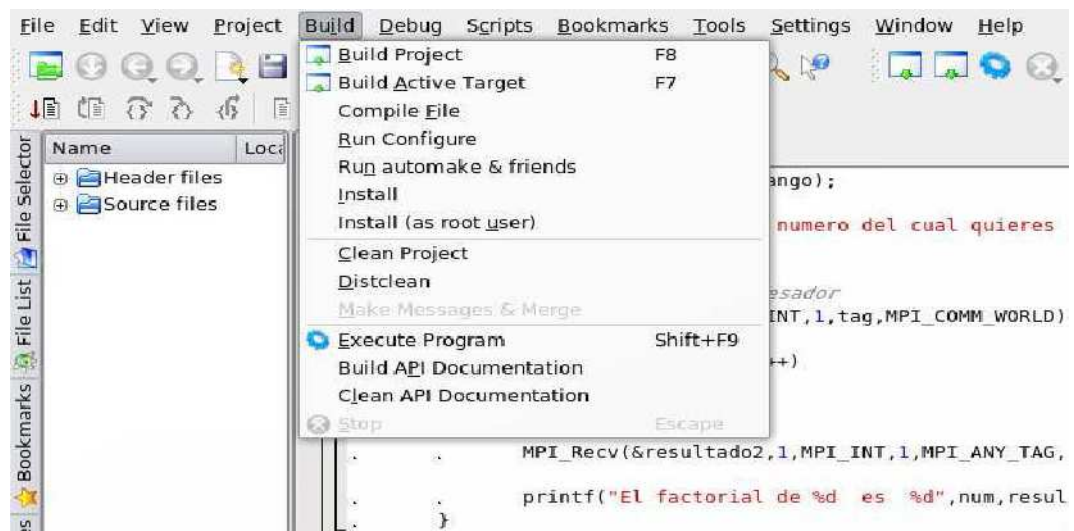
Menú Build -> Install

Menú Build -> Run Automake & Friends

Menú Build -> Run Configure

Para compilar se dá click en *Menu Build -> Build Project* y posteriormente en *Menu Build -> Build Active Target*.

Para ejecutar tu programa click en *Menu Build -> Execute Program*



Después de ejecutar tu programa debe aparecer una pantalla como la siguiente:

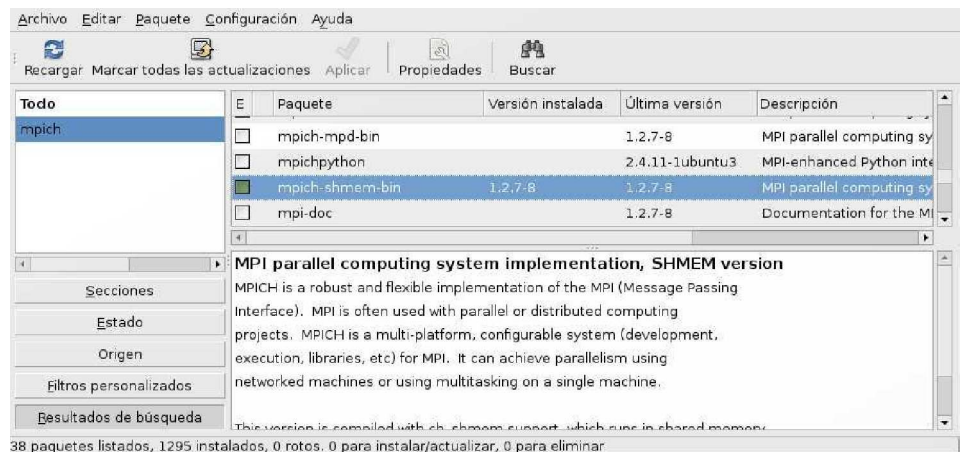


5. Configurando openMP para Fortran

5.1 Paquetes a instalar

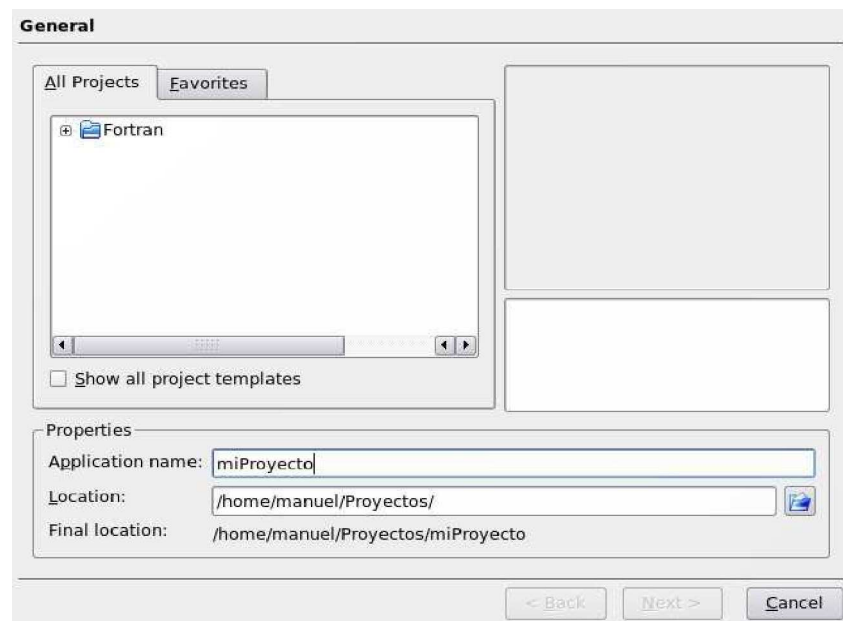
Debe instalar los siguientes paquetes, sólo si no instaló los necesarios para usar MPI.

- binutils
- automake
- autoconf
- g++
- gcc
- gfortran
- m4
- libtool
- konsole
- kdevelop



5.2 Creando un programa en Kdevelop

Menu Project -> New Project. Elegir de tipo Fortran y escribir el nombre del proyecto a crear y elegir la ubicación del mismo.

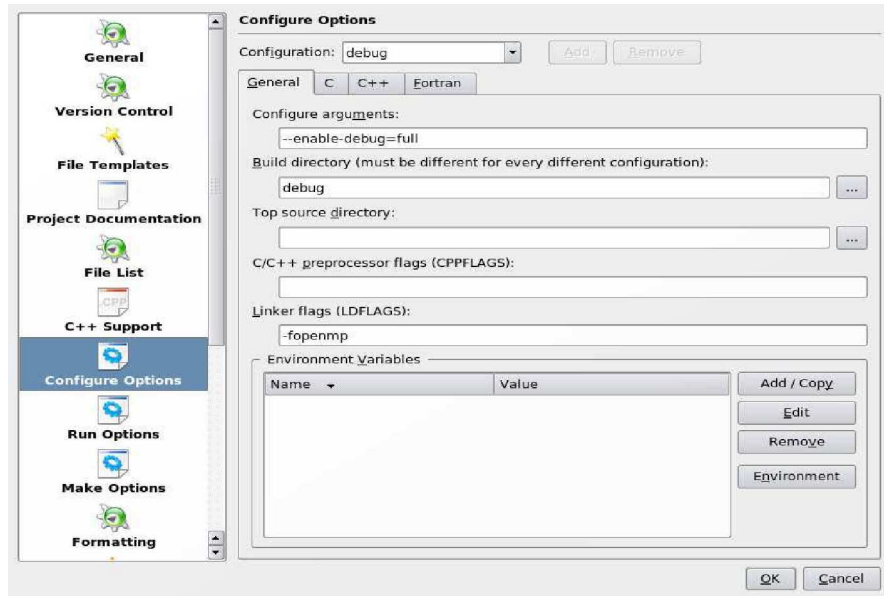


5. Configurando openMP para Fortran

Tec. Manuel Antonio Ortiz Montalvo

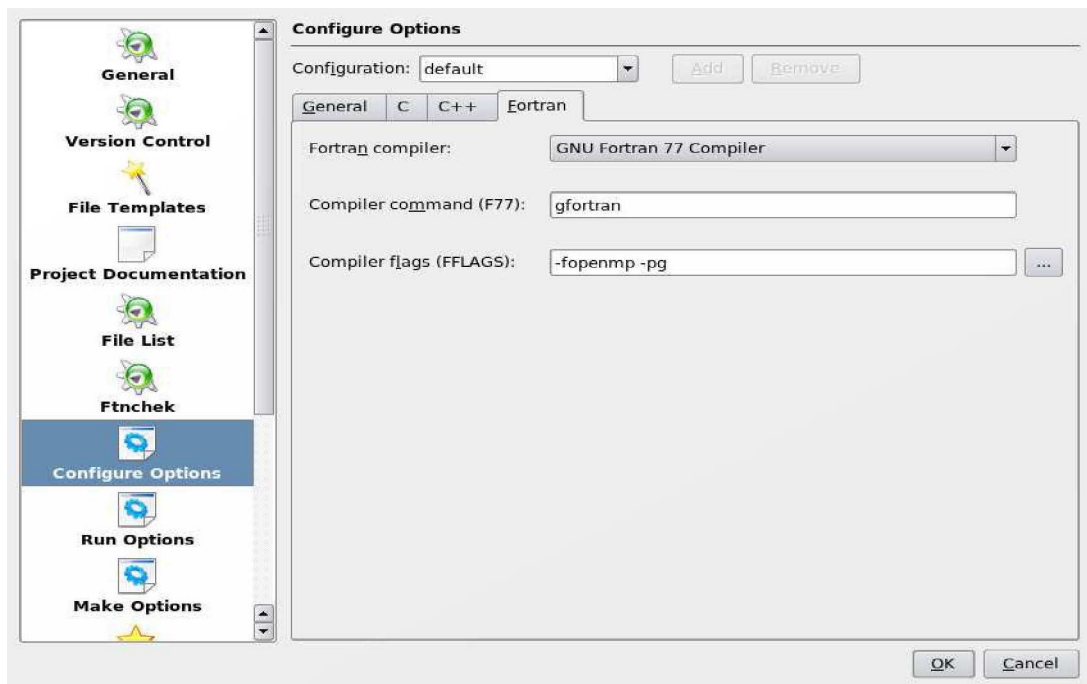
Para configurar openMP, debe dar click en Menú Project -> Project Options -> Configure Options, y en la pestaña general debe configurar la siguiente opción:

Linker Flags (LDFLAGS): -fopenmp



Y en la pestaña Fortran, debe configurar las siguientes opciones

Fortran Compiler: GNU Fortran 77 Compiler
Compiler Command (f77): gfortran
Compiler Flags (FFLAGS): -fopenmp -pg



Posteriormente aceptar los cambios dando click en OK.

5. Configurando openMP para Fortran

Tec. Manuel Antonio Ortiz Montalvo

Escribir el siguiente código de ejemplo, para demostrar como funciona openMP. Nótese que debe incluir la libreria `omp_lib.h`, como se muestra en la siguiente figura.

```
openmpfortran.f
c
c Ejemplo de openMP
c
program hello
include "omp_lib.h"
integer hilo,noHilos,OMP_GET_NUM_THREADS,OMP_GET_THREAD_NUM
call OMP_SET_NUM_THREADS(3)
!$OMP PARALLEL PRIVATE(hilo) SHARED(noHilos)
hilo=OMP_GET_THREAD_NUM()
noHilos=OMP_GET_NUM_THREADS()
print *, 'Hilo ',hilo,' De ',noHilos
!$OMP END PARALLEL
stop
end
```

Antes de compilar por primera vez, debes dar click en las siguientes opciones:

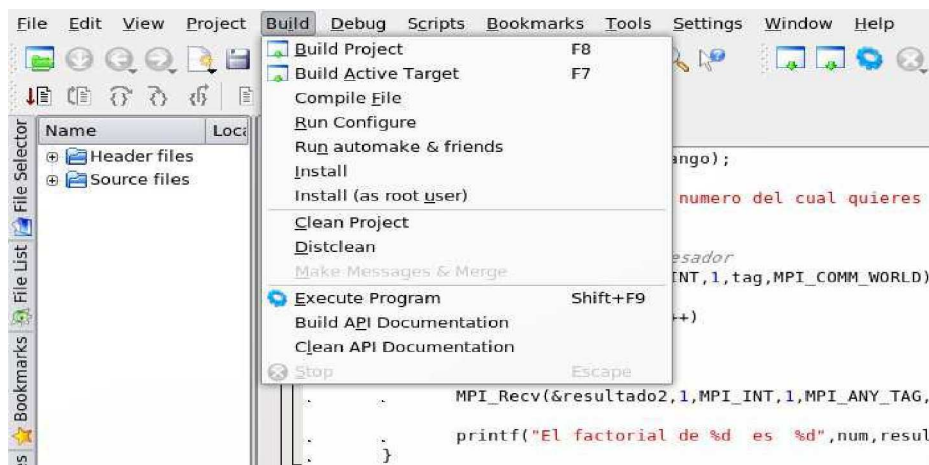
Menú *Build* -> *Install*

Menú *Build* -> *Run Automake & Friends*

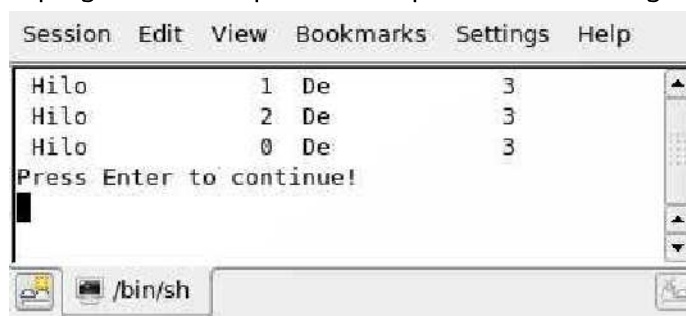
Menú *Build* -> *Run Configure*

Para compilar se dá click en Menu *Build* -> *Build Project* y posteriormente en Menu *Build* -> *Build Active Target*.

Para ejecutar tu programa click en Menu *Build* -> *Execute Program*



Después de ejecutar tu programa debe aparecer una pantalla como la siguiente:

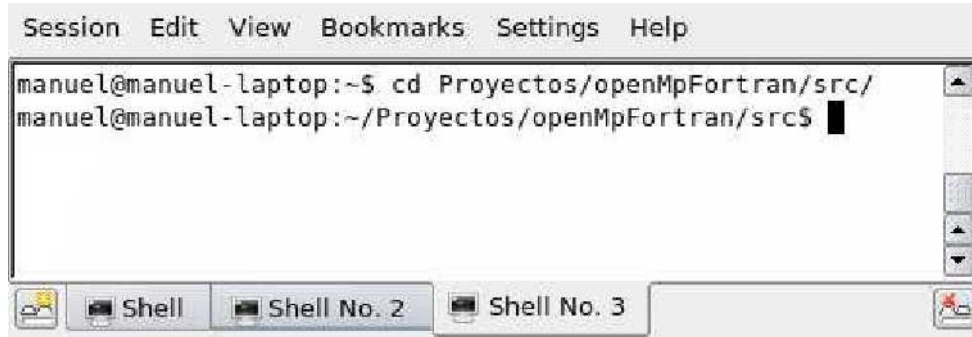


5. Configurando openMP para Fortran

Tec. Manuel Antonio Ortiz Montalvo

5.3 Compilando y ejecutando desde la línea de comandos

Para compilar desde la línea de comandos, primero debes ubicarte en la carpeta que contiene el código fuente.



```
Session Edit View Bookmarks Settings Help
manuel@manuel-laptop:~$ cd Proyectos/openMpFortran/src/
manuel@manuel-laptop:~/Proyectos/openMpFortran/src$
```

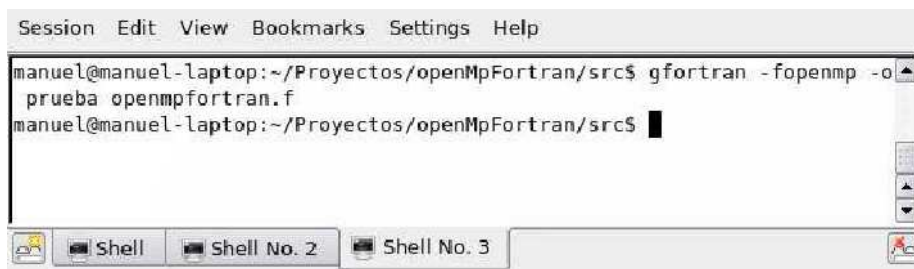
Posteriormente ejecutar el siguiente comando para realizar la compilación

gfortran -pg -fopenmp -o programaEjecutable codigoFuente.f

Dónde:

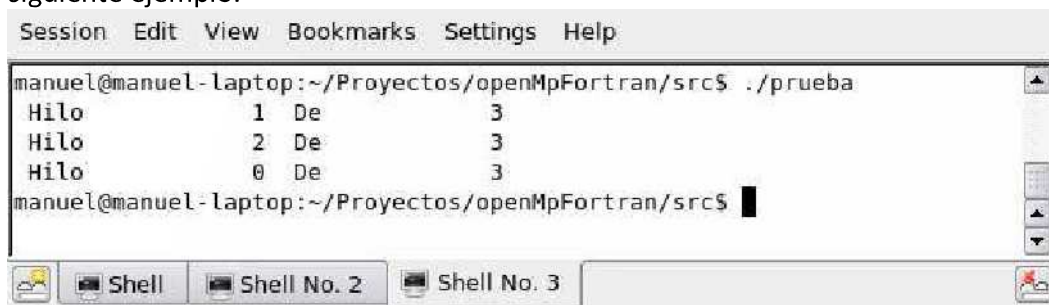
-pg Activa la opción de generar el Profile
-fopenmp Activa la opción de crear ambiente paralelo
-o Genera el archivo objeto
programaEjecutable Nombre del archivo ejecutable, que generará la compilación
codigoFuente.c Nombre de tu archivo fuente

A continuación se muestra un ejemplo:



```
Session Edit View Bookmarks Settings Help
manuel@manuel-laptop:~/Proyectos/openMpFortran/src$ gfortran -pg -fopenmp -o
prueba openmpfortran.f
manuel@manuel-laptop:~/Proyectos/openMpFortran/src$
```

Y ya para ejecutar tu programa, sólo debes escribir el nombre de tu ejecutable que se acaba de generar, antecedido por los signos “ ./ “, y tu programa debe empezar a ejecutarse, como muestra el siguiente ejemplo:



```
Session Edit View Bookmarks Settings Help
manuel@manuel-laptop:~/Proyectos/openMpFortran/src$ ./prueba
Hilo            1 De            3
Hilo            2 De            3
Hilo            0 De            3
manuel@manuel-laptop:~/Proyectos/openMpFortran/src$
```

6. Creando el profile para Fortran

Si se desea hacer un análisis de la ejecución del código, los tiempos que demoró la ejecución de cada método, y las veces que se llamo cada uno, entre mucha otra información, se puede crear el profile.

Para crearlo, primeramente el programa se tiene que compilar activando la bandera pg:

```
gfortran -pg -openmp -o programa archivoFuente.f
```



```
Session Edit View Bookmarks Settings Help
manuel@manuel-laptop:~/Proyectos/openMpFortran/src$ gfortran -pg -fopenmp
p -o prueba openmpfortran.f
manuel@manuel-laptop:~/Proyectos/openMpFortran/src$
```

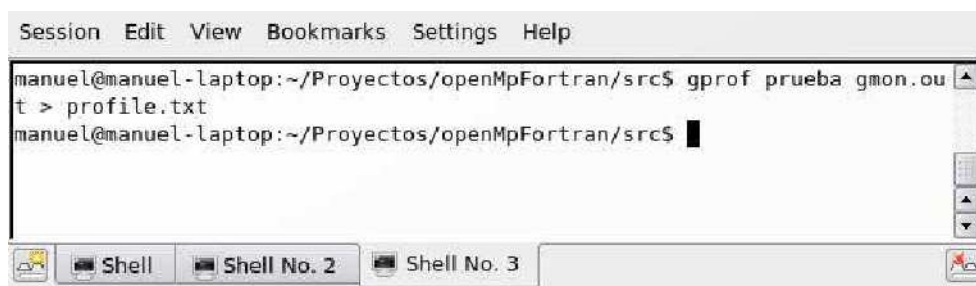
Si compiló correctamente, se tuvo que generar un archivo llamado gmon.out en la carpeta de tu código fuente.

Posteriormente, se crea tu archivo profile, de la siguiente manera:

```
gprof programa gmon.out > archivo.txt
```

Donde:

programa: Es el nombre del ejecutable que se generó durante la compilación
gmon.out Archivo que se generó durante la compilación con la bandera pg
archivo.txt Nombre del archivo que generará con la información del profile



```
Session Edit View Bookmarks Settings Help
manuel@manuel-laptop:~/Proyectos/openMpFortran/src$ gprof prueba gmon.out
t > profile.txt
manuel@manuel-laptop:~/Proyectos/openMpFortran/src$
```

Para visualizar el archivo que generó, sólo tienes que escribir el comando:

```
edit nombreArchivo
```

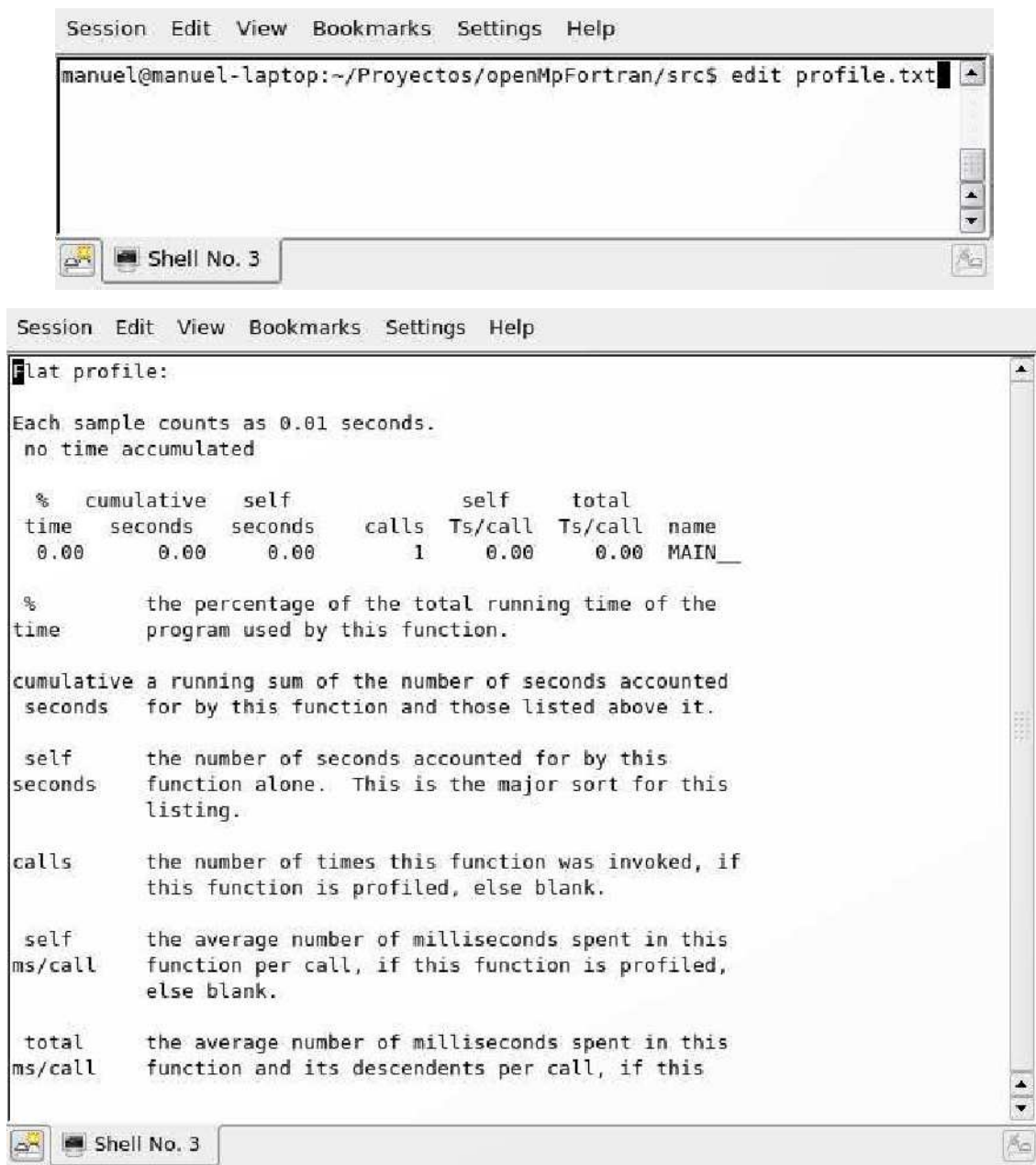
Donde:

nombreArchivo: Es el nombre del archivo a visualizar, que es el que generamos con el comando gprof

6. Creando el profile para Fortran

Tec. Manuel Antonio Ortiz Montalvo

A continuación se muestra un ejemplo de la ejecución del comando, y el resultado del mismo.



The first screenshot shows a terminal window with the command `manuel@manuel-laptop:~/Proyectos/openMpFortran/src$ edit profile.txt` entered. The second screenshot shows the output of the command, which is a flat profile. The output includes a table with columns for time, cumulative seconds, self seconds, calls, Ts/call, and name. The table shows a single entry for the MAIN function with a value of 1 for calls and 0.00 for Ts/call. Below the table, there are several lines of text explaining the meaning of the columns and the profile itself.

```
Session Edit View Bookmarks Settings Help
manuel@manuel-laptop:~/Proyectos/openMpFortran/src$ edit profile.txt
Shell No. 3
```

```
Session Edit View Bookmarks Settings Help
Flat profile:
Each sample counts as 0.01 seconds.
no time accumulated

%   cumulative   self             self           total
time  seconds  seconds   calls   Ts/call  Ts/call  name
0.00    0.00    0.00        1      0.00    0.00  MAIN__

%           the percentage of the total running time of the
time        program used by this function.

cumulative  a running sum of the number of seconds accounted
seconds     for by this function and those listed above it.

self        the number of seconds accounted for by this
seconds     function alone. This is the major sort for this
           listing.

calls       the number of times this function was invoked, if
           this function is profiled, else blank.

self        the average number of milliseconds spent in this
ms/call     function per call, if this function is profiled,
           else blank.

total       the average number of milliseconds spent in this
ms/call     function and its descendents per call, if this
```

Shell No. 3