

# ACARUS

# CAR



## **Responsable :**

María del Carmen Heras Sánchez

## **Asesores Técnicos :**

Daniel Mendoza Camacho

Yessica Vidal Quintanar

# *Día 3:*

- ✓ **Compilación**
- ✓ **Ejecución**

# *Cluster* Choya

# Compilaciones Choya

- **Código Serial**

Lenguaje	Compilador	Comando
C	Intel	icc

- **Código Paralelo**

Lenguaje	Compilador	Versión	Comando
C	MPI	mpich	mpicc

# Hola Mundo - Serial - C

- **Crear archivo hola\_mundo\_serial.c**

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

```
int main()
```

```
{
```

```
    printf("Hola mundo \n");
```

```
    return 0;
```

```
}
```

# Hola Mundo – Serial - C

- `srun --pty -p batch /bin/bash -i`
- **Compilar `hola_mundo_serial.c` usando Intel**  
`module load intel/13.1.1`  
`icc hola_mundo_serial.c -o hola_mundo_serial-ci`
- **Ejecutar `hola_mundo_serial-ci`**  
`./hola_mundo_serial-ci`

# Hola Mundo – Paralelo – C

- **Crear archivo hola\_mundo\_mpi.c**

```
#include <stdio.h>
#include <mpi.h>

int main (argc, argv)
    int argc;
    char *argv[];
{
    int rank, size;

    MPI_Init (&argc, &argv);
    MPI_Comm_rank (MPI_COMM_WORLD, &rank);
    MPI_Comm_size (MPI_COMM_WORLD, &size);
    printf( "Hola Mundo %d of %d\n", rank, size );
    MPI_Finalize();
    return 0;
}
```

# Hola Mundo – Paralelo - C

- **Compilar código MPI usando mpich**  
module load gnu/mpich-3.1.2  
mpicc hola\_mundo\_mpi.c -o hola\_mundo\_mpi-c
- **Ejecutar hola\_mundo\_mpi-c**  
mpirun -np 4 ./hola\_mundo\_mpi-c



*Cluster*

**Ocotillo**

# Compilaciones Ocotillo

- **Código Serial**

Lenguaje	Compilador	Comando
C	Open64	openc
C	Intel	icc
Fortran	Open64	openf90
Fortran	Intel	ifort

- **Código Paralelo**

Lenguaje	Compilador	Versión	Comando
C	MPI	openmpi	mpicc
Fortran	MPI	openmpi	mpif90

# Hola Mundo - Serial - C

- Crear archivo hola\_mundo\_serial.c

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

```
int main()
```

```
{
```

```
    printf("Hola mundo \n");
```

```
    return 0;
```

```
}
```

# Hola Mundo – Serial - C

- **Lanzar un job interactivo**

```
qsub -I -1 nodes=1:ppn=1
```

- **Compilar hola\_mundo\_serial.c usando open64**

```
module load open64/5.0
```

```
opencpp hola_mundo_serial.c -o hola_mundo_serial-co
```

- **Ejecutar hola\_mundo-co**

```
./hola_mundo-co
```

# Hola Mundo – Serial - C

- **Compilar hola\_mundo\_serial.c usando Intel**  
module load intel/parallel\_studio-2013  
icc hola\_mundo\_serial.c -o hola\_mundo\_serial-ci
- **Ejecutar hola\_mundo\_serial-ci**  
./hola\_mundo\_serial-ci

# Hola Mundo - Serial - Fortran

- **Crear archivo hola\_mundo\_serialF.f90**

```
PROGRAM hola_mundo
    PRINT *, "Hola mundo"
END PROGRAM hola_mundo
```

# Hola Mundo – Serial - Fortran

- **Compilar hola\_mundo\_serialF.f90 usando open64**  
module load open64/5.0  
openf90 hola\_mundo\_serialF.f90 -o hola\_mundo\_serialF-fo
- **Ejecutar hola\_mundo\_serialF-fo**  
./hola\_mundo\_serialF-fo

# Hola Mundo – Serial - Fortran

- **Compilar hola\_mundo\_serialF.f90 usando Intel**  
module load intel/parallel\_studio-2013  
ifort hola\_mundo\_serialF.f90 -o hola\_mundo\_serialF-fi
- **Ejecutar hola\_mundo\_serialF-fi**  
./hola\_mundo\_serialF-fi



# Hola Mundo – Paralelo – C

- **Crear archivo hola\_mundo\_mpi.c**

```
#include <stdio.h>
#include <mpi.h>

int main (argc, argv)
    int argc;
    char *argv[];
{
    int rank, size;

    MPI_Init (&argc, &argv);
    MPI_Comm_rank (MPI_COMM_WORLD, &rank);
    MPI_Comm_size (MPI_COMM_WORLD, &size);
    printf( "Hola Mundo del proceso %d of %d\n", rank, size );
    MPI_Finalize();
    return 0;
}
```

# Hola Mundo – Paralelo - C

- **Compilar código MPI usando openmpi**  
module load gnu/openmpi-1.4.3  
mpicc hola\_mundo\_mpi.c -o hola\_mundo\_mpi-c
- **Ejecutar hola\_mundo\_mpi-c**  
mpirun -np 8 ./hola\_mundo\_mpi-c

# Hola Mundo – Paralelo - Fortran

- **Crear archivo hola\_mundo\_mpi.f90**

```
program hola_mundo_mpi
  include 'mpif.h'
  integer rank, size, ierror, tag, status(MPI_STATUS_SIZE)

  call MPI_INIT(ierror)
  call MPI_COMM_SIZE(MPI_COMM_WORLD, size, ierror)
  call MPI_COMM_RANK(MPI_COMM_WORLD, rank, ierror)
  print*, 'node', rank, ': Hola Mundo'
  call MPI_FINALIZE(ierror)
end
```

# Hola Mundo – Paralelo - Fortran

- **Compilar código MPI usando openmpi**

```
module load gnu/openmpi-1.4.3
```

```
mpif90 hola_mundo_mpi.f90 -o hola_mundo_mpi-f
```

- **Ejecutar hola\_mundo\_mpi-c**

```
mpirun -np 8 ./hola_mundo_mpi-f
```

# Ejecuciones

- **Serial**

```
./hola_mundo_serial-co
```

```
./hola_mundo_serial-ci
```

```
./hola_mundo_serialF-fo
```

```
./hola_mundo_serialF-fi
```

- **Paralelo**

```
mpirun -np 8 ./hola_mundo_mpi-c
```

```
mpirun -np 8 ./hola_mundo_mpi-f
```