

# Examples

about how to use argv and argc parameters

## Sum.c

```
int main(int argc, char *argv[]){  
int N, Strategy;  
...  
N=atoi(argv[1]);  
Strategy=atoi(argv[2]);  
...  
return 0;  
}
```

**atoi** is a function to transform a char/string variable in an integer one

## In the pbs file

```
/usr/lib64/openmpi/1.4-gcc/bin/mpicc -o $PBS_O_WORKDIR/Sum $PBS_O_WORKDIR/Sum.c
```

```
/usr/lib64/openmpi/1.4-gcc/bin/mpiexec -machinefile $PBS_NODEFILE -np $NCPU $PBS_O_WORKDIR/Sum 1000000 3
```

(Equivalent to sequential run from the workdir `./Sum 1000000 3` )

In this case, it will be  
argc=3  
argv[0]="Sum"  
argv[1]="1000000" (N)  
argv[2]="3" (Strategy)

Actual values of argv and argc during THIS execution

## Sum.c

```
int main(int argc, char *argv[]){  
int N, Strategy;  
...  
N=atoi(argv[1]);  
Strategy=atoi(argv[2]);  
...  
return 0;  
}
```

## In the pbs file

```
/usr/lib64/openmpi/1.4-gcc/bin/mpicc -o $PBS_O_WORKDIR/Sum $PBS_O_WORKDIR/Sum.c
```

```
/usr/lib64/openmpi/1.4-gcc/bin/mpiexec -machinefile $PBS_NODEFILE -np $NCPU $PBS_O_WORKDIR/Sum 100 1
```

(Equivalent to sequential run from the workdir `./Sum 100 1` )

In this case, it will be

argc=3

argv[0]="Sum"

argv[1]="100" (N)

argv[2]="1" (Strategy)

Actual values of argv and argc during THIS execution

## Sum.c

```
int main(int argc, char *argv[]){  
int N, Strategy;  
...  
N=atoi(argv[1]);  
Strategy=atoi(argv[2]);  
...  
return 0;  
}
```

## In the pbs file

```
/usr/lib64/openmpi/1.4-gcc/bin/mpicc -o $PBS_O_WORKDIR/Sum $PBS_O_WORKDIR/Sum.c
```

```
/usr/lib64/openmpi/1.4-gcc/bin/mpiexec -machinefile $PBS_NODEFILE -np $NCPU $PBS_O_WORKDIR/Sum 10000 2
```

(Equivalent to sequential run from the workdir `./Sum 10000 2` )

In this case, it will be

argc=3

argv[0]="Sum"

argv[1]="10000" (N)

argv[2]="2" (Strategy)

Actual values of argv and argc during THIS execution

## Sum.c

```
int main(int argc, char *argv[]){  
int N, Strategy;  
...  
N=atoi(argv[1]);  
Strategy=atoi(argv[2]);  
...  
return 0;  
}
```

## In the pbs file

```
/usr/lib64/openmpi/1.4-gcc/bin/mpicc -o $PBS_O_WORKDIR/Sum $PBS_O_WORKDIR/Sum.c
```

```
/usr/lib64/openmpi/1.4-gcc/bin/mpiexec -machinefile $PBS_NODEFILE -np $NCPU $PBS_O_WORKDIR/Sum 1000000000 2
```

(Equivalent to sequential run from the workdir `./Sum 1000000000 2` )

In this case, it will be  
argc=3  
argv[0]="Sum"  
argv[1]="1000000000" (N)  
argv[2]="2" (Strategy)

Actual values of argv and argc during THIS execution

## Sum.c

```
int main(int argc, char *argv[]){
int N, Strategy;
...
N=atoi(argv[1]);
Strategy=atoi(argv[2]);
if ((Strategy==3)&&(argc>3))
    Printer=atoi(argv[3]);
...
return 0;
}
```

If the user decides for the 3° strategy, I check if she wants also to decide which one has to print the result

## In the pbs file

```
/usr/lib64/openmpi/1.4-gcc/bin/mpicc -o $PBS_O_WORKDIR/Sum $PBS_O_WORKDIR/Sum.c
```

```
/usr/lib64/openmpi/1.4-gcc/bin/mpiexec -machinefile $PBS_NODEFILE -np $NCPU $PBS_O_WORKDIR/Sum 1000000000 3 5
```

(Equivalent to sequential run from the workdir `./Sum 1000000000 3 5` )

In this case, it will be  
argc=4  
argv[0]="Sum"  
argv[1]="1000000000" (N)  
argv[2]="3" (Strategy)  
argv[3]="5" (Printer)

Actual values of argv and argc during THIS execution