

Bash tricks

Get disk usage

```
df -h
```

Will give you something like this:

Sys. de fichiers	Taille	Utilisé	Dispo	Uti%	Monté sur
/dev/sde1	110G	8,1G	97G	8%	/
devtmpfs	126G	0	126G	0%	/dev
tmpfs	126G	0	126G	0%	/dev/shm
tmpfs	126G	920K	126G	1%	/run
tmpfs	126G	0	126G	0%	/sys/fs/cgroup
tmpfs	126G	4,0K	126G	1%	/tmp
/dev/mapper/ddf1_DATA2	15T	7,7G	14T	1%	/home

Get CPU and RAM usage

```
htop
```

Will give you something like this:

htop.png

To get more information about htop see [here](#)

Start a Job without a queue

All process started from ssh are terminated when you close the ssh connection, even if you fork them (./app &). They are closed because when a process is closed the system send the SIGTERM signal to all its children, it's done to avoid zombies process on a machine. To keep your task alive when you disconnect from ssh, you should use screen ([tutorial](#)), it will block the SIGTERM signal. To start your application with screen:

```
screen # to start screen
./your_app #to start your application or any other command
           #type ' Ctrl-A' d to leave screen with your application running in background
```

To reconnect to your previous session:

```
screen -ls # to list running sessions
screen -r 33287.pts-36.bender # to reconnect to 33287.pts-36.bender session
exit # to close your screen session
```

Start a mpi code:

Please note that the default mpi distribution on [Bender](#) /[Flexo](#) /[Jakolass](#) is a custom one compiled with intel compiler and libraries, to use the gcc one you should call mpi(cc/f90/run/...) with the full path /usr/lib64/openmpi/bin/mpi(cc/f90/run/...). To run a software compiled with mpi (mpic/cxx/fortran), uses mpirun with -np to set the number of mpi processes you want to start.

```
mpirun -np 32 /path_to_myapp/myapp
#to run myapp with 32 MPI processes
```

Start a openMP code:

When you run a software compiled with openMP library, you can tune the number of openMP threads your code will run. To do this you just have to set the OMP_NUM_THREADS environment variable.

```
export OMP_NUM_THREADS=32
/path_to_myapp/myapp
#to run myapp with 32 openMP threads
```

Abort a Job:

To stop a running job you can use the kill command in different ways

- You have just one Job running on the machine and you know its name

```
killall TheNameOfTheJob
```

- You have multiple jobs running, you have first to get the Process ID of the job

```
ps ax # will list the running processes  
# You can filter, for example if you have started your job with mpi  
ps ax | grep mpi
```

The first number you get is the Process ID

To kill it

```
kill -9 123456 # if 123456 is your Process ID
```

Files

htop.png	133 KB	06/11/2013	Alexis Jeandet
----------	--------	------------	----------------