

# Very Quick Start

Running Examples

Timothy H. Kaiser, Ph.D.

[tkaiser@mines.edu](mailto:tkaiser@mines.edu)



# Purpose

- Make sure that your environment is working on Mio - RA is the same
- Actually do something very quickly
- Introduce you to the documentation

# We will compile and run 2 examples

- Download a tar file from the documentation page
- Unpack the tar file
- Run make
- You then run the examples

# Web Page

<http://geco.mines.edu/guide>

<http://geco.mines.edu/guide/quickstart>

# Getting On

- All access to RA & MIO is via ssh
- Should use password less ssh with a nontrivial pass phrase
- This can be set up so that you only need to type a pass phrase once a day
- See: <http://geco.mines.edu/ssh/>
- Today we will use passwords



# Getting On

- On Windows use PuTTY
  - All Programs
    - All Applications
      - PuTTY
        - PuTTY
- On OSX - use Terminal
  - Applications
    - Utilities
      - Terminal

# PuTTY - the Movie



# Shortcut Setup

<http://inside.mines.edu/mio/page3.html>

Add the following to you .bashrc file and log out/in

```
if [ -f /usr/local/bin/setup/setup ]; then source /usr/local/bin/setup/setup intel ; fi
```

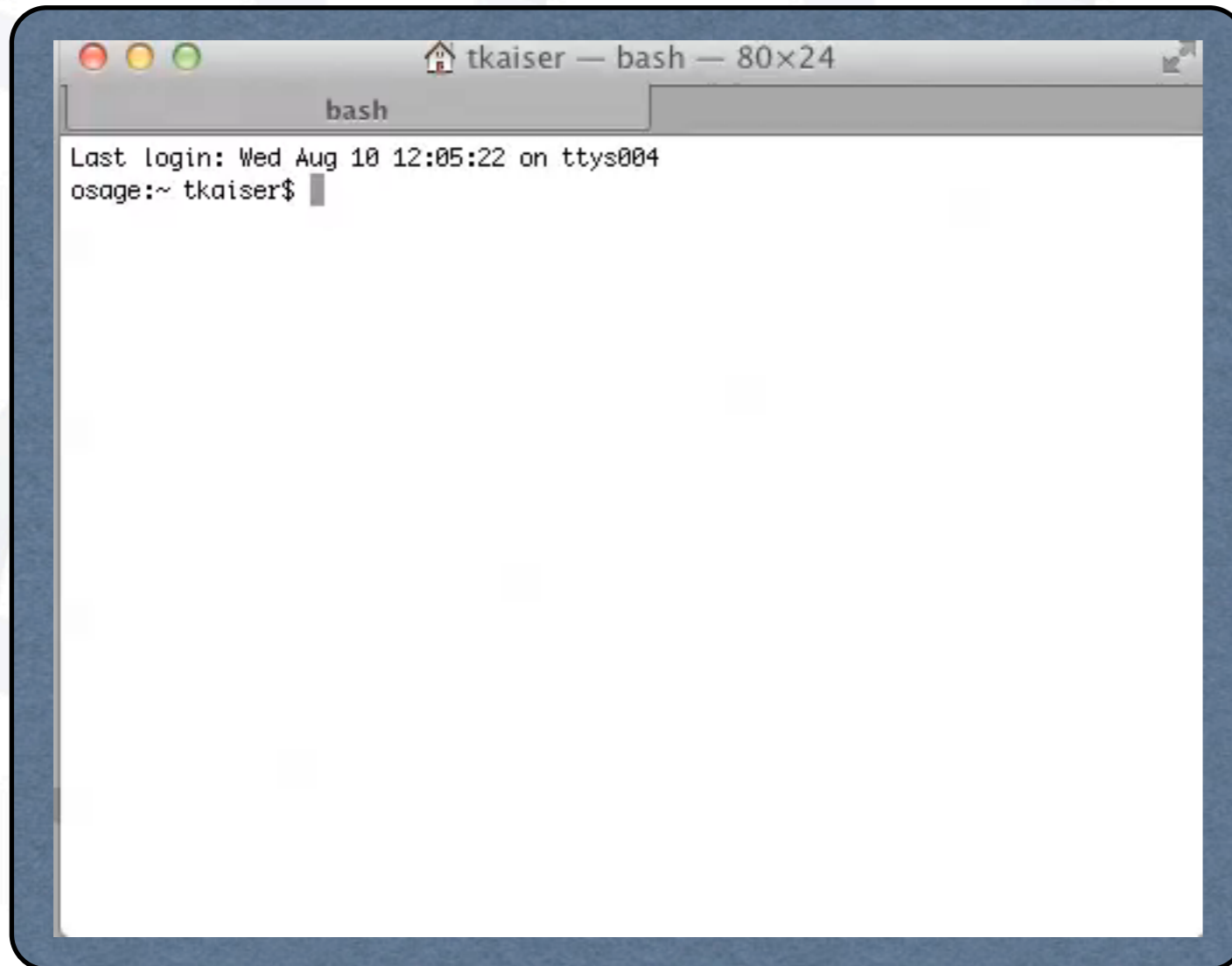
- Compilers
  - Intel I I.x
  - AMD
  - Portland Group
  - NAG
- Openmpi parallel MPI environment
  - MPI compilers
  - MPI run commands
- Python 2.6.5 and 3.1.2

Should give you: 

```
[tkaiser@mio ~]$ which mpicc  
/opt/lib/openmpi/1.4.2/intel/11.1/bin/mpicc  
[tkaiser@mio ~]$
```



# After you are on...



```
tkaiser — bash — 80x24
bash
Last login: Wed Aug 10 12:05:22 on ttys004
osage:~ tkaiser$
```

- Check on MPI version
- Check on \$SCRATCH directory
- Check on \$DATA directory
  - We will build our application in \$DATA

# Procedure

What you need type (Copy/Paste) is in red.

(1) On "mio.mines.edu", create a new empty directory.

```
mkdir guide
```

(2) Go to the directory.

```
cd guide
```

(3) Download the tar file

```
wget -o wget.out http://geco.mines.edu/guide/guideFiles/quick.tar
```

(3b) You might want to do an "ls" to make sure the file "quick.tar" actually downloaded.

(4) Untar the file.

```
tar -xf quick.tar
```

(4b) You can do an "ls" to list the files c\_ex00.c, f\_ex00.f, makefile, and runfile.

(5) Make the executables

```
make
```

(6) Submit your job

```
msub runfile -l flags=ADVRES:work2.9189
```

# makefile

```
default: exec
```

```
exec:f_ex00 c_ex00
```

```
f_ex00:f_ex00.f  
    mpif90 f_ex00.f -o f_ex00
```

```
c_ex00:c_ex00.c  
    mpicc c_ex00.c -o c_ex00
```

# Fortran Example f\_ex00.f

```
!*****
! This is a simple hello world program. Each processor
! prints out its rank and total number of processors
! in the current MPI run.
!*****

program hello
include "mpif.h"
character (len=MPI_MAX_PROCESSOR_NAME):: myname
call MPI_INIT( ierr )
call MPI_COMM_RANK( MPI_COMM_WORLD, myid, ierr )
call MPI_COMM_SIZE( MPI_COMM_WORLD, numprocs, ierr )
call MPI_Get_processor_name(myname,mylen,ierr)
write(*,*)"Hello from ",myid," of ",numprocs," on ",trim(myname)
call MPI_FINALIZE(ierr)
stop
end
```



# C Example c\_ex00.c

```
#include <stdio.h>
#include <stdlib.h>
#include <mpi.h>
#include <math.h>
/*****
This is a simple hello world program. Each processor prints out
it's rank and the size of the current MPI run (Total number of
processors).
*****/
int main(argc,argv)
int argc;
char *argv[];
{
    int myid, numprocs,mylen;
    char myname[MPI_MAX_PROCESSOR_NAME];

    MPI_Init(&argc,&argv);
    MPI_Comm_size(MPI_COMM_WORLD,&numprocs);
    MPI_Comm_rank(MPI_COMM_WORLD,&myid);
    MPI_Get_processor_name(myname,&mylen);

    /* print out my rank and this run's PE size*/
    printf("Hello from %d of %d on %s\n",myid,numprocs,myname);

    MPI_Finalize();
}
```

# PBS Runscript

```
#!/bin/bash
#PBS -l nodes=1:ppn=8
#PBS -l walltime=02:00:00
#PBS -N testIO
##PBS -o out.$PBS_JOBID
##PBS -e err.$PBS_JOBID
#PBS -o stdout
#PBS -e stderr
#PBS -r n
#PBS -V
#-----

cd $PBS_O_WORKDIR

#save a nicely sorted list of nodes
sort -u $PBS_NODEFILE > mynodes.$PBS_JOBID

echo "running the C example"
mpiexec -n 8 c_ex00
echo "running the Fortran example"
mpiexec -n 8 f_ex00
```

# Our Directory after the run

```
[tkaiser@ra ~/guide]$ ls -lt
total 732
-rw----- 1 tkaiser tkaiser  958 Jun 17 14:54 stdout
-rw-rw-r-- 1 tkaiser tkaiser   18 Jun 17 14:54 mynodes.9637.ra.local
-rw----- 1 tkaiser tkaiser    0 Jun 17 14:54 stderr
-rwxrwxr-x 1 tkaiser tkaiser 57210 Jun 17 14:54 c_ex00
-rwxrwxr-x 1 tkaiser tkaiser 648987 Jun 17 14:54 f_ex00
-rw-rw-r-- 1 tkaiser tkaiser  409 Jun 17 14:54 wget.out
-rw-rw-r-- 1 tkaiser tkaiser 10240 Jun 17 14:53 quick.tar
-rw-r--r-- 1 tkaiser tkaiser  235 Jun 17 14:53 makefile
-rw-r--r-- 1 tkaiser tkaiser  458 Jun 17 14:37 runfile
-rw-r--r-- 1 tkaiser tkaiser  673 Apr  9 11:58 f_ex00.f
-rw-r--r-- 1 tkaiser tkaiser  758 Apr  9 11:46 c_ex00.c
[tkaiser@ra ~/guide]$
```

# Typical output

```
[tkaiser@ra ~/guide]$ cat stdout
```

```
Warning: no access to tty (Bad file descriptor).
```

```
Thus no job control in this shell.
```

```
running the C example
```

```
Hello from 0 of 8 on compute-5-7.local
```

```
Hello from 2 of 8 on compute-5-7.local
```

```
Hello from 3 of 8 on compute-5-7.local
```

```
Hello from 5 of 8 on compute-5-7.local
```

```
Hello from 6 of 8 on compute-5-7.local
```

```
Hello from 1 of 8 on compute-5-7.local
```

```
Hello from 4 of 8 on compute-5-7.local
```

```
Hello from 7 of 8 on compute-5-7.local
```

```
running the Fortran example
```

```
Hello from      2 of      8 on compute-5-7.local
```

```
Hello from      4 of      8 on compute-5-7.local
```

```
Hello from      1 of      8 on compute-5-7.local
```

```
Hello from      0 of      8 on compute-5-7.local
```

```
Hello from      6 of      8 on compute-5-7.local
```

```
Hello from      7 of      8 on compute-5-7.local
```

```
Hello from      3 of      8 on compute-5-7.local
```

```
Hello from      5 of      8 on compute-5-7.local
```

```
[tkaiser@ra ~/guide]$
```



# To get all of the examples on Ra:

```
mkdir class1
cd class1
wget http://geco.mines.edu/workshop/class2/examples/examples.tgz
tar -xzf examples.tgz
```

This gives you something like:

```
[tkaiser@ra ~/class1]$ ls
dotar  examples.tgz  friday  hybrid  index.html  mpi  openmp
stommel
```