Supercomputers Architecture

Hands on 1

November 24th, 2015

Constantino Gomez

Albert Segura

Cristobal Ortega

Exercise 1: Send an email to support@bsc.es ✓
Exercise 2: Change your password. ✓
Exercise 3: List the files in your home directory.

```
~> ls
bin
~>
```

Exercise 4: Using linux commands (e.g. cd, ls, …) determine how many users and apps there are at MN III .

```
~> ls -l /gpfs/apps/MN3 | wc -l
280
```

#Real homes are on subdirectories:

```
~> ls -l /gpfs/home/*/ | grep "^d" | wc -l
3238
~> ls -l /gpfs/projects/ | grep "^d" | wc -l
518
~> ls -l /gpfs/scratch/ | grep "^d" | wc -l
537
```

Apps: 280
Users: 3238 grouped in 537 prefixes

Exercise 5: Inspect the content of your local hard drive. Has the previous user removed his/her data?

```
#Check the variable
~> echo $TMPDIR
/scratch/tmp
```

```
#There are a lot of files
~> ls $TMPDIR | wc -l
139948
```

```
#None of us
~> ls -ltr $TMPDIR | grep "sam14021"
~>
```

Exercise 6: Use one of these tools for transferring a file between your local machine and Marenostrum.

```
$ scp hello.c sam14021@mn3.bsc.es:/home/sam14/sam14021/hello.c
```

Exercise 7: Inspect the "/gpfs/archive" directory. Describe what happened. #WRONG

```
~>  ls /gpfs/archive
ls: cannot access /gpfs/archive: No such file or directory
```

Exercise 8: Check your assigned quota.

```
~> quota
```

No limits for apps, home, projects or scracth

Exercise 9: Create a "Hello World" program, compile it and run it.

```
~> vim hello_world.c
~> gcc hello_world.c -o hello_world
     hello_world.c: In function 'main':
     hello_world.c:3: warning: return type of 'main' is not 'int'
~> ./hello_world
Hello World
```

Exercise 10 – OPTIONAL: Implement a serial matrix-vector multiplication C program (mat_vect_mult.c) using one-dimensional arrays to store the vectors and the matrix. As an input the mat_vect_mult program will receive the dimensions of the matrix (m = number of rows, n= number of columns). The values of the matrix can be obtained directly by the rand() function. The output will be the product vector y = Ax.

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

void print_matrix(int* matrix, int nrows, int ncolumns) {
     printf("===MATRIX===\n");
     for(int i = 0; i < nrows; ++i){
               for(int j = 0; j < ncolumns; ++j)
                    printf("%i ",matrix[i*nrows+j]);
               printf("\n");
```

```
	}

}

void print_vector(int* vector, int ncolumns) {
	printf("===VECTOR===\n");
	for(int j = 0; j < ncolumns; ++j)
			printf("%i ",vector[j]);
	printf("\n");

}

int main(int argc, char **argv) {
	int nrows = atoi(argv[1]);
	int ncolumns = atoi(argv[2]);

	int *matrix = malloc(nrows*ncolumns*sizeof(int));
	int *vector = malloc(ncolumns*sizeof(int));

	srand(time(NULL));

	for(int i = 0; i < nrows; ++i)
			for(int j = 0; j < ncolumns; ++j){
				matrix[i*nrows+j] = rand();
				vector[j] = 1;
			}


	print_matrix( matrix, nrows, ncolumns);
	print_vector( vector, ncolumns);

	for(int i = 0; i < nrows; ++i)
			for(int j = 0; j < ncolumns; ++j)
				matrix[i*nrows+j] *= vector[j];

	print_matrix( matrix, nrows, ncolumns);

}
```