HW01

Chien-Ming Chen (cchen556@wisc.edu)

2. Write a single line of bash for each of the following (you can use pipes and/or redirects). Pretend that any files or directories mentioned exist (don’t turn them in).

a) Print the current working directory

```
pwd
```

b) Change directories into a directory called somedir

```
cd somedir
```

c) List the names of all files in the current directory

```
ls
```

d) (Optional) Write a for loop which prints each integer 0 to 10.

```
for((i=1;i<=10;i++)); do echo $i; done
```

e) (Optional) Print the names of all files ending in .txt in the current directory or any of its subtree

```
find . -name '*.txt'
```

f) (Optional) Print the last 2 lines of a plain text file in the current directory called sometext.txt

```
tail -2 sometext.txt
```

g) (Optional) Print the entire contents of a file in the current directory called sometext.txt

```
cat sometext.txt
```

3. Using Euler and the module command, answer the following questions.

a) Are there any modules loaded (module list) when you log in?

No

b) What version (version number) of gcc is available to you without loading any modules?

```
$ gcc –version
gcc (GCC) 8.3.1 20190507 (Red Hat 8.3.1-4)
```

c) List all gcc modules available on Euler.

```
$ module avail gcc
------------------------------------------------ /usr/local/share/modulefiles ------------------------------------------------
gcc/0_cuda/5.3.0  gcc/4.4.7  gcc/4.x.x  gcc/5.3.0  gcc/6.1.x  gcc/7.1.x  gcc/8.1.x      gcc/9.2.0
gcc/0_cuda/6.4.0  gcc/4.8.4  gcc/5.1.0  gcc/5.3.x  gcc/6.4.0  gcc/7.2.0  gcc/8.2.0      gcc/acc/6.3.0
gcc/0_cuda/7      gcc/4.8.x  gcc/5.1.x  gcc/5.5.0  gcc/6.4.x  gcc/7.3.0  gcc/8.2.0-el7  gcc/acc/8.1.1
gcc/0_cuda/7.1.0  gcc/4.9.2  gcc/5.2.0  gcc/5.x.x  gcc/6.x.x  gcc/7.x.x  gcc/8.2.x      gcc/latest
gcc/0_cuda/8.2.0  gcc/4.9.x  gcc/5.2.x  gcc/6.1.0  gcc/7.1.0  gcc/8.1.0  gcc/8.x.x
gcc/next-recommended gcc/recommended
```

d) Which gcc module is loaded when you run module load gcc and what version number of gcc is loaded by that module?

```
$ gcc (GCC) 8.2.0
```

e) List one other piece of software that you know that has a module on Euler and one sentence about what it does. (If you aren’t familiar with any of the other software, look one up and write a sentence about it.)

llvm

a set of compiler and toolchain which can be used to develop a front end for any programming language and a back end for any instruction set architecture.

4. Write a bash script called task4.sh with a Slurm header which asks for

- 2 CPU cores
- A job name of FirstSlurm
- An output file called FirstSlurm.out
- An error file called FirstSlurm.err

and runs a single command to print the hostname of the machine (compute node) running the job. This job should be submittable by running sbatch task4.sh on the head node.

task4.sh is updated on Canvas

```
#!/usr/bin/env bash

#SBATCH -p wacc

#SBATCH -t 0-00:30:00

#SBATCH -J FirstSlurm

#SBATCH -o FirstSlurm.out -e FirstSlurm.err

#SBATCH -c 2

echo hostname = $(hostname)
```

5. Research some useful Slurm tools (one sentence responses):

a) Explain what SLURM SUBMIT DIR is in the environment of a running Slurm job.

The directory from which **sbatch** was invoked, or, if applicable, the directory specified by the -D, --chdir option.

b) In what directory does a Slurm job on euler begin execution. You should run some jobs to check this?

The same directory from which **sbatch** was invoked to run the Slurm job

c) Explain what SLURM JOB ID is in the environment of a running Slurm job.

The ID of the Slurm job allocation

d) Explain what the following script header line specifies: #SBATCH --array=0-9

submit a job array with index values between 0 and 9

e) Explain what SLURM ARRAY TASK ID is in the environment of a running Slurm job with the header from part (d).

SLURM ARRAY TASK ID will be set to the job array index value, i.e. a value between 0 to 9

f) Explain what the following script header line specifies: #SBATCH --gres=gpu:1

allocate a single gpu

6. Write a C++ program called task6.cpp which takes a command line argument N, and prints out each integer from 0 to N (including 0 and N) separated by spaces on a single line ending in a newline.

- Compile command: g++ task6.cpp -Wall -O3 -o task6
- Run command: ./task6 N
- Expected output (followed by newline): 0 1 2 3 ⋯ N
- Expected output for N = 6 (followed by newline): 0 1 2 3 4 5 6

task6.cpp is updated on Canvas

```
#include <iostream>
#include <string>
int main(int argc, char* argv[])
{
  if (argc < 2) {
    std::cerr << "error!! please check the argument" << std::endl;
    return 1;
  }
  int N = std::stoi(argv[1],nullptr,0);
  for(int i = 0; i <= N; i++)
    std::cout << i << " ";
  std::cout << std::endl;
  return 0;
}
```