Exercise 9: simple bash script

Write a **bash script** (call it **`blast_script.sh`**) to launch a BLAST search using the input data downloaded previously and the command from the lecture

```
#!/bin/bash

blastx -db ./databases/swissprot -query seq_tst.fa –out hits.txt >& blast.out
```

Make sure the script is executable (**`chmod u+x blast_script.sh`**)

In another terminal, start the top program (**`top -u myID`**)

Run the script you just created in the background (**`./blast_script.sh &`**). Observe the **`top`** output in the other window. How much CPU and memory is the run taking?

Exercise 10: running BLAST on two CPUs

Modify the script **`blast_script.sh`** you created previously to

1. run BLAST using **two** CPU threads (hint: use the **`–num_threads`** option)
2. accept the name of the query FASTA file to be processed as an **argument**

```
#!/bin/bash

MYQUERY=$1

blastx -num_threads 2 -db ./databases/swissprot -query $MYQUERY -o ${MYQUERY}.out >& ${MYQUERY}.log
```

Run the script you just modified in the background, specifying the input file as an argument

```
./blast_script.sh seq_test.fa &
```

Observe the **`top`** output in the other window. How much CPU and memory is the run taking now?

Exercise 11: Multiple independent tasks run sequentially

Copy compressed files **`AAA_1.gz`**, **`AAA_2.gz`**, **`AAA_3.gz`** from **`/shared_data/Linux_workshop/auxfiles`**

Write a bash script (call it **`gunzip_script.sh`**) to un-compress the three files using **`gunzip`** tool. Script can be written in many equivalent ways, e.g.:

```
#!/bin/bash

gunzip AAA_1.gz
gunzip AAA_2.gz
gunzip AAA_3.gz
```

```
#!/bin/bash

for i in AAA_1 AAA_2 AAA_3
do
    gunzip ${i}.gz
done
```

```
#!/bin/bash

for i in {1..3}
do
    gunzip AAA_${i}.gz
done
```

Run the script (type **`./gunzip_script.sh`**). Observe the top output in the other terminal window.
How many running process do you see?

Exercise 12: Multiple independent tasks run in parallel

Copy files **BBB_1**, **BBB_2**, **BBB_3** from **/shared_data/Linux_workshop/auxfiles**

Make a copy of script **gunzip_script.sh**, calling it **gzip_script.sh**. Edit this new script and change command **gunzip** to **gzip**. Also, add & after each **gzip** command to send it to the background:

```
#!/bin/bash

gzip BBB_1 &
gzip BBB_2 &
gzip BBB_3 &
```

```
#!/bin/bash

for i in BBB_1 BBB_2 BBB_3
do
    gzip ${i} &
done
```

```
#!/bin/bash

for i in {1..3}
do
    gzip BBB_${i} &
done
```

Run the script (type **./gzip_script.sh**)

Why did the script “end immediately”?
Observe the **top** output in the other terminal window. How many running process do you see?

Exercise 13: Large number of independent tasks run in parallel on limited CPUs

We will use a “fake” program (written in perl), called **`rndwait.pl`**, located in **`/shared_data/Linux_workshop/auxfiles`**

Copy the executable to your working directory

The program takes one parameter (an integer). Try running it and see what output it gives

```
./rndwait.pl 20
```

Objective: run the command above **100 times**, with the values of argument 1, 2, 3, …, 100, using **3 CPUs**

Solution: prepare a file (call it **`taskfile`**), containing 100 commands, one per line. You can use a script similar to

```
#!/bin/bash

rm –f taskfile
for i in {1..100}
do
  echo "./rndwait.pl $i >& out.$i‟ >> taskfile
done
```

Examine the resulting **`taskfile`**

Exercise 13: Large number of independent tasks run in parallel on limited CPUs - continued

Start running the 100 commands from your **`tasklist`** using at most 3 CPUs at a time

```
/programs/bin/perlscripts/perl_fork_univ.pl   tasklist   3  >& log &
```

Monitor the output from the top command in the other window. What processes do you see? Which ones are consuming CPU and how many of them?

List the content of your scratch directory. Do you see the output files generated by **`rndwait.pl`** ?