

Building GotoBLAS

In this exercise, we'll use the same conventions and commands as in all previous. You should refer back to the previous Exercise descriptions for details on various Unix commands.

1. Open up a web browser and go to your favorite search engine (e.g., Google).

2. Search for:

```
GotoBLAS
```

3. Go to the Texas Advanced Computing Center (TACC) software website.

4. Go to the list of software available for download.

5. Click on GotoBLAS2.

6. Click on Downloads.

7. Follow the links to download the GotoBLAS source code.

8. Upload the source code to the computer you want to install it on.

9. Log in to the computer that you want to build GotoBLAS on.

10. Go into your `NCSIPARI2011_exercises` directory:

```
% cd ~/NCSIPARI2011_exercises
```

11. Create a directory to do this build in:

```
% mkdir GotoBLAS
```

12. Go into that directory:

```
% cd GotoBLAS
```

13. Move the compressed tar file to you uploaded to that directory; for example:

```
% mv ~/GotoBLAS2-1.13.tar.gz ~/NCSIPARI2011_exercises/GotoBLAS/
```

14. “Untar” the compressed tar file:

```
% tar zxvf GotoBLAS2-1.13.tar.gz
```

This may take a few minutes.

15. Go into the newly created GotoBLAS directory:

```
% cd GotoBLAS2
```

16. Determine your current working directory:

```
% pwd
```

17. Build GotoBLAS via this command:

```
% nohup make FC=gfortran >& make_output.txt &
```

This will take several minutes. (The bit about `gfortran` – which is the GNU implementation of Fortran – is because GotoBLAS defaults to commercial compilers if it finds any, but we're sticking to a pure GNU build, for simplicity.)

NOTE: `nohup` means “Even if I get logged out, keep going;” the `>&` means “redirect stdout and stderr to the following file;” the ampersand `&` at the end means “do this in background.”

18. Check that the output from the `make` command (`make_output.txt`), specifically the end of the file, shows that you built GotoBLAS successfully.