

# ENEE 785

## How to install and run the OCS code

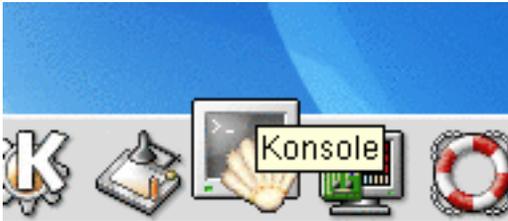
John Zweck and Brian Marks

March 8, 2003

### **1 Logging into and using the ECS lab machines**

Follow these instructions to log into the lab machines and use them to download and install the OCS software:

1. Choose one of the Dell machines in the ECS labs.
2. Assuming the machine you choose is running Windows 2000, it needs to be rebooted into Linux for your work for this class. Do this by pressing Ctrl+Alt+Del, then click on “Yes”, then click “Shutdown”, and then choose “Restart” from the menu. When you click OK, the machine will shutdown and reboot.
3. After the Dell splash screen, you will have a choice of Windows or Linux. Press the down-arrow to get Linux and press Enter.
4. You may see a prompt for “LILO Boot:”. Press Enter or wait a few seconds, and the machine will boot into Linux.
5. Choose KDE from the pull-down menu and try logging in with your username and password. KDE is a windowing environment on Unix machines that resembles Microsoft Windows and is convenient to use. You may go through a series of menus to set up KDE. The default choices for setting up KDE are probably fine.
6. Click on the console icon at the bottom of the screen:



A new shell window should appear.

7. Type “hostname” at the prompt and take note of what it says. Example:

```
[marks@ecs021pc07-lx ~]$ hostname  
ecs021pc07-lx.ucslab.umbc.edu
```

8. Type the following to allow windows from malkhut to appear on your computer:

```
[marks@ecs021pc07-lx ~]$ xhost +malkhut. engr. umbc. edu  
malkhut. engr. umbc. edu being added to access control list
```

9. To get into malkhut, type

```
[marks@ecs021pc07-lx ~]$ ssh marks@malkhut. engr. umbc. edu  
marks@malkhut. engr. umbc. edu's password:  
Last login: Mon Jan 27 16:40:09 2003 from 130.85.112.174  
  
[marks@malkhut ~]$
```

However, of course, you would substitute “marks” for your username.

10. Now that you’re logged into malkhut, type the following to cause malkhut to display windows on your local (lab) machine’s screen:

```
[marks@malkhut ~]$ setenv DISPLAY ecs021pc07-lx.ucslab.umbc.edu:0
```

**Note:** The name of the machine you are on in the lab is whatever the response was from the “hostname” command — `ecs021pc07-lx.ucslab.umbc.edu` is just an example.

**Note #2:** The `:0` at the end of the hostname is **important and meaningful**. Your windows will not be displayed properly if you forget this!

11. Now if you type “`netscape http://www.photonics.umbc.edu/ENEE785/ &`” on malkhut, you should see a Netscape window appear in a few seconds with the class homepage on it.

## 2 Installing and compiling the code

Follow these instructions to download the OCS code and input parameter files

1. Download the files `ocs.tar.gz` and `expts.tar.gz` from the internal course web page at <http://malkhut.engr.umbc.edu/internal/ENEE785/> into your malkhut home directory
2. Use the follow sequence of unix commands in your malkhut account to install and compile the ocs code
  - `cd ~`
  - `mkdir ENEE785`
  - `cd ENEE785`
  - `mkdir ocs`
  - `mkdir expts`
  - `cd ocs`
  - `mv ~/ocs.tar.gz .`
  - `gunzip ocs.tar.gz`
  - `tar -xvf ocs.tar`
  - `cd src`
  - `make depend`
  - `make upmyapp`
3. At this point you will have compiled the ocs code. The binary file will be in the directory `~/ENEE785/ocs/bin` and is called `myapp`.
4. Before running the code you first need to set up the input parameter files by doing
  - `cd ~/ENEE785/expts`
  - `mv ~/expts.tar.gz .`
  - `gunzip expts.tar.gz`
  - `tar -xvf expts.tar`
  - `cd BackToBackNoiseFreeSystem`
  - `ls`
5. Now you will see the input parameter files with suffixes `".in"`

### 3 Running the code

To actually run the example application code given in `~/ENEE785/ocs/src/MyApp.cc` you should do the following

- `cd ~/ENEE785/expts/BackToBackNoiseFreeSystem`
- `~/ENEE785/ocs/bin/myapp` [*This command runs the executable*]

After less than a minute you should see “THE END” written to the screen. If you do an “ls” you will find several files with the prefix “T1”. These are the output files. You can read the log file `T1.log` and you can use Matlab to view the output data in the `.dat` files.

To plot the data in Matlab, you can download the file `less_simple_plot.m` off the class website and put it in the `~/ENEE785/expts/BackToBackNoiseFreeSystem` directory. Run Matlab from the command line by typing

```
[marks@malkhut ~]$ matlab -nojvm
```

The `-nojvm` option causes Matlab to use fewer computer resources on malkhut, meaning that you will not slow down your classmates as much! At the Matlab command line prompt (`>>`), type

```
>> less_simple_plot('T1_tInOpt.dat')
```

You will soon see the return-to-zero bit string that was generated at the beginning part of the `myapp` code. If you type in Matlab

```
>> less_simple_plot('T1_fInOpt.dat')
```

you will see the Fourier spectrum of this bit string. Similarly, you can plot the output data by running

```
>> less_simple_plot('T1_tReceivedCurrent.dat')
```

in Matlab. The `T1_tReceivedCurrent.dat` file has a column for the absolute value of the received current as well as a column for the real part of the electrical current after the received signal has been passed through a 5th-order Bessel filter. The `less_simple_plot` program plots both of these curves on the same plot. You will see the differences between the two curves if you zoom in around the “valleys” in a 101 pattern.

The reason the files are prefixed with “T1” is that in `Startup.in` I set the `JobName` to be `T1`. If you want to run again with different parameters and not write over the previous job you should change the `JobName` say to “T2”.

## 4 A Handy Hint

I hate typing the full path `~/ENEE785/ocs/bin/myapp` of the executable when I run the code. So I add the directory `~/ENEE785/ocs/bin` to my path in my `.tcshrc` file. To do that:

1. `cd ~`
2. Edit the file `.tcshrc` and add the line  

```
set path = ( $path ~/ENEE785/ocs/bin )
```
3. Then do the commands:
  - `source .tcshrc`
4. In the future to run the code just do:
  - `cd ~/ENEE785/expts/BackToBackNoiseFreeSystem`
  - `myapp` [*This command runs the executable*]

## 5 Compiling and running the code after you have modified it

At some point you will start modifying the code yourself. How do you recompile and run the modified code?

1. First try to recompile the code
  - `cd ~/ENEE785/ocs/src`
  - `make myapp` [*This compiles the code*]
2. You may get error messages. If so attempt to fix you bugs and recompile with `make myapp` until the code has compiled
3. To replace the old executable in `~/ENEE785/ocs/bin` with the new one you have just made do `make upmyapp`. Here the *up* stands for update!
4. Test your changes by running the new executable as explained above.
5. Technical Point: If you ever add more `#include` directives in the header or source files you will need to do `make depend` before `make myapp`.