TÉCNICO
LISBOA

Circuit Theory and Electronics Fundamentals

Lecture 6: Introduction to the lab classes online – 2nd Part

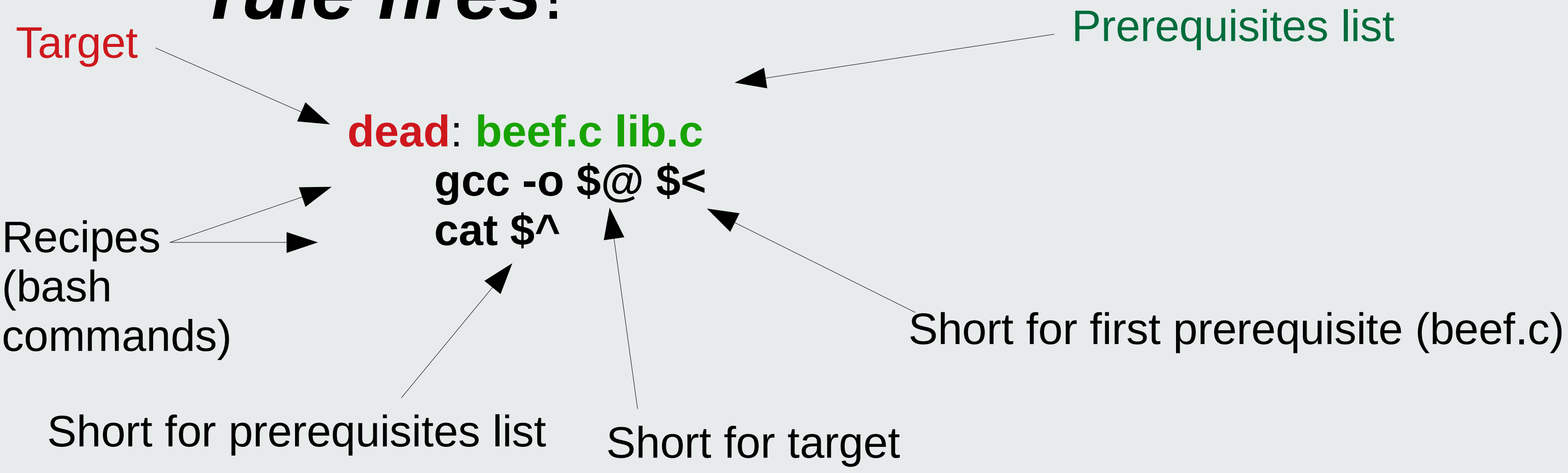
- Make
- Example lab assignement T0
  - Top Makefile
  - Octave script, log and Makefile
  - Ngspice script, log and Makefile
  - Latex files and Makefile
  - LibreOffice drawings

The Make utility program

- Make is a program that automates the process of building programs (compiling)
- Created by Steward Feldman in 1976
  - Also wrote the first Fortran 77 compiler
  - Part of the Bell Labs team who created the Unix operating system, the precursor of Linux (Unix for the masses)
- Make is still one of the best build automation tools to date!

The Make utility program

- Make programs are called *Makefiles*
- A *Makefile* is based on rules.
- If target ***older*** than any prerequisite ***rule fires***!



Target

Prerequisites list

```
dead: beef.c lib.c
      gcc -o $@ $<
      cat $^
```

Recipes (bash commands)

Short for first prerequisite (beef.c)

Short for prerequisites list

Short for target

The example lab assignment t0

- ***Makefile***: top make program
- ***mat/t0.m***: Octave script for running theoretical calculations
- **sim/t0.net**: Ngspice script for running circuit simulations
- **doc**: directory for building the project report
  - Makefile: make program for building the document
  - report.tex: top Latex file which includes the different document sections
  - Frontcover.tex: document cover
  - Intro.tex: introduction section
  - Analysis.tex: theoretical calculations section
  - Simulation.tex: circuit simulation section
  - Conclusion.tex: conclusion section
  - rc.odg: Libreoffice drawing to include in the text

Top Makefile

```
# --------------------------------------------------------
# type "make" command in the Linux terminal create report.pdf
#
# type "make clean" to delete all generated files
# --------------------------------------------------------
all:
        make -C mat
        make -C sim
        make -C doc
        cp doc/report.pdf .
clean:
        make -C mat clean
        make -C sim clean
        make -C doc clean

.PHONY: all clean
```

target

No pre-requisites

Bash commands:
Call Makefile in folders mat, sim and doc
Copy report.pdf from doc to current directory

target

No pre-requisites

Bash commands:
Call Makefile in folders mat, sim and doc to remove generated files

Phony targets list: a phony target is executed UNCONDITIONALLY

The octave script (symbolic)

```
close all
clear all

%%EXAMPLE SYMBOLIC COMPUTATIONS

pkg load symbolic

syms t
syms R
syms C
syms vi(t)
syms vo(t)
syms i(t)

i(t)=C*diff(vo,t)

printf("\n\nKVL equation:\n");

vi(t) = R*i(t)+vo(t)
```

Symbolic variables and functions

Solving symbolic equations with respect to some variable

```
syms vo_n(t) %natural solution
syms vo_f(t) %forced solution

v(t) = vo_n(t) + vo_f(t)

syms A
syms wn

vi(t) = 0 %no excitation
i_n(t) = C*diff(vo_n, t)

vo_n(t) = A*exp(wn*t)

R*i_n(t)+vo_n(t) == 0

R*C*wn*vo_n(t)+vo_n(t) == 0

R*C*wn+1==0

solve(ans, wn)
```

The octave script (numeric)

Response of an RC series circuit

```
R=1e3 %Ohm
C=100e-9 %F

f = 1000 %Hz
w = 2*pi*f; %rad/s

%time axis: 0 to 10ms with 1us steps
t=0:1e-6:10e-3; %s

Zc = 1/(j*w*C)
Cgain = Zc/(R+Zc)
Gain = abs(Cgain)
Phase = angle(Cgain)

vi = 1*cos(w*t);
vo = Gain*cos(w*t+Phase);
```

Plotting and file printing

```
hf = figure ();
plot (t*1000, vi, "g");
hold on;
plot (t*1000, vo, "b");

xlabel ("t[ms]");
ylabel ("vi(t), vo(t) [V]");
print (hf, "forced.eps", "-depsc");
```

The mat makefile (runs octave)

```
octave.log: rc.m
	octave $< > $@
```

Runs script rc.m in octave and redirects standard output to octave.log

```
clean:
	@rm -f octave.log octave-workspace *.eps *~
```

Removes generated files

Declares clean as phony target

```
.PHONY: clean
```

The octave log

Shown directly from terminal….

- The log is formed by the runtime messages that are output to the terminal:
  - Errors and warnings
  - Results from commands not terminated by ‘;’
  - User messages: not currently used but can be used in the future to create .tex files and figures used by Latex

The ngspice script

Shown directly from terminal….

- Script has two parts:
  - Circuit description
  - Simulator control
- Description part – describe the circuit to be simulated
- Control part – control the simulator to simulate the circuit statically, in the *time-domain* and in the *frequency-domain*
- Static or *Operating Point* analysis requires input values
- Time-domain analysis requires input time functions
- Frequency-domain analysis requires input frequency functions

The ngspice log

Shown directly from terminal….

- The log is formed by the messages that are output to the terminal:
  - Errors and warnings
  - User messages:
    - used to output figure names to be converted the from ps to pdf; the pdf figures are included in the Latex document
    - Used to output data tables which are processed to obtain .tex files that are included in the Latex document

The ngspice Makefile

Shown directly from terminal….

- Runs the ngspice script on ngspice to produce the ngspice log
- Processes the ngspice log to
  - Find all .ps figures produced by ngspice and convert them to pdf to be included in the Latex document
  - Find the data tables produced by ngspice and convert them to a .tex file to be included in the Latex document

The Latex document

Shown directly from terminal….

- Enables structured and ***hierarchical*** document development
- Hierarchy top: ***report.tex***
  - Includes other document .tex files using the \input{file} directive
- ***report.tex***: top file
  - ***frontcover.tex:*** document cover: title, authors, organization
  - ***intro.tex***: introduction
  - ***analysis.tex***: theorectical analysis
  - ***simulation.tex***: simulation analysis
  - ***conclusion.tex***: conclusion
- Figure inclusion using package *graphicx* (shown in simulation.tex)
- Table inclusion using just the \input{file} directive (shown in simulation.tex)

The doc Makefile

Shown directly from terminal….

- Runs the LibreOffice in batch mode to convert .odg drawings into pdf files (can you run Microsoft Office in batch mode?)
- Unified drawing environment for the whole organization: draw, share, edit any file
- Runs Latex to produce the document report.pdf
- Opens report.pdf file for human inspection