

RML

setting and debugging crashcourse

Adrian Pop
PELAB/IDA/LiU
adrpo@ida.liu.se

- Things you should already know a bit of
 - Unix-like environments
 - Makefile writing
 - Emacs

- rml = your rml directory
- rml/elisp/rml-mode.el
- /home/you/.emacs
- add these in .emacs

```
(setq load-path (cons "rml/elisp/" load-path))  
(setq auto-mode-alist  
      (cons `("\\.rml$" . rml-mode)  
            auto-mode-alist))  
(global-font-lock-mode)  
(autoload `rml-mode "rml-mode" "RML editing mode." t)
```

loading rml debugger mode in Emacs

- rml = your rml directory
- rml/elisp/rml-db.el
- /home/you/.emacs
- add this in .emacs

```
(load-file "rml/elisp/rml-db.el")
```

building your rml project in debug mode

- use rmlc flags: -g when compiling .rml files
- link with:
 - flags -lrml_g instead of -lrml
 - you may need also -lfl and -lreadline
- if you use rmlc to compile/link the C files then you don't need to bother about linking because it does it automatically
- the program you build in debug mode *has the debugger included in it.*

using Emacs to debug your program

- write a small wrapper for your program and name it `run-program.sh`

```
#!/bin/sh
```

```
./your_program its_parameters
```

```
#-- end here
```

- don't forget to make it executable:

```
chmod +x run-program.sh
```

- then run emacs in that directory:
 - `emacs ./your_rml_file.rml`
- in Emacs: M=ALT or M=Esc then x
 - M-x rmlldb <ENTER>

using RMLDataTree browser

- the browser resides in rml/RMLDataTree
- run it with:

```
java -cp . RMLDataTree path_to_rml_files
```

- Short documentation `rml/doc/DEBUG.txt`
- Small Website with RML info:
 - <http://www.ida.liu.se/~adrpo/teaching/sstpl/>
 - you can find also this presentation there
- Contact, if you get stuck:
 - Adrian Pop, adrpo@ida.liu.se