CS2LAP: Logic and Prolog 2000/2001
Using Linux and SWI-Prolog

SWI-Prolog, a free Prolog system developed at the University of Amsterdam, is installed under Linux at the Department of Computer Science. You may also install SWI-Prolog on your computer at home (Unix, Linux, or Windows). A link to the SWI-Prolog download page may be found on the course web site.

Starting Linux. After booting, choose *Linux* and log in with your usual user name and password. Now you have to start a window manager, which is usually done by typing `startx`. At the Department this command might instead display a choice of other commands to start a particular window manager. Choose one.

To start working you need to open a *terminal*. For some window managers there will be a menu or a button on the desktop which lets you do this. In any case, if you have at least one terminal open you can start a new one by typing `xterm &` in the first one.

Important Linux commands. Under Linux, most things are done by typing commands into terminals. The following table summarises some of the most important ones.

Important Unix/Linux commands	
`mkdir` *directory-name*	create the new directory *directory-name*
`cd` *directory-name*	change to directory *directory-name* (`..` is always one directory down, ~ is your home directory)
`ls`	list contents of current directory
`cp` *source destination*	copy a file from *source* to *destination*
`mv` *source destination*	move a file from *source* to *destination*
`pwd`	print working directory on screen
`man` *command*	view manual pages for *command*

You should start by creating a directory for your Prolog programs and then change to that directory (`>` denotes the Linux prompt – it will probably look different on your machine):

```
> mkdir cs2lap
> cd cs2lap
```

Editing files. Files (including Prolog programs) are edited using an editor, for example *Emacs*. The command to start Emacs is `emacs`. If you wish, you can give a file name as an argument, or you may open a file (or several files) from the menus once Emacs is running. Example:

```
> emacs hello.pl &
```

The `&` at the end tells the system to run the `emacs`-command as a background process, that means the prompt will be free to accept further commands. As a general rule, use `&` whenever the process you intend to start involves opening a new window (e.g. `emacs`, `xterm`, `netscape`, but not `ls` etc.).

Running Prolog. Type `pl` to start the Prolog interpreter. (This is an example for a command where you should definitely not use `&` or things will get confusing.)

```
> pl
Welcome to SWI-Prolog (Version 3.3.9)
Copyright (c) 1990-2000 University of Amsterdam.
Copy policy: GPL-2 (see www.gnu.org)

For help, use ?- help(Topic). or ?- apropos(Word).

?-
```

The Prolog prompt is `?-`. Make sure you're not typing Linux commands into Prolog or Prolog queries into Linux – it just won't work. You can start straight away by typing in simple queries. Example:

```
?- example( _, dog, dog) = example( cat, Animal, _).
Animal = dog
Yes
```

As this get's boring pretty soon you should start writing your own programs. Do that using Emacs (see previous page), but remember to either quit Prolog first or run Emacs from another terminal. To *compile* your program in Prolog use the `consult/1` predicate. Suppose you have written a Prolog program and saved it (very important!) in the file `hello.pl`.

```
?- consult( 'hello.pl').
% hello.pl compiled 0.00 sec, 716 bytes
Yes
```

Those are single quotes. Always. If your file name ends in `.pl` (which it should), you may also type `consult( hello).` instead. Please remember that everything you type into the Prolog interpreter has to end with a dot. Also remember there are no blank characters allowed between the functor of a predicate (here `consult`) and the following opening bracket.

Exit Prolog by using the `halt/0` predicate, that is by typing `halt.`

Logging out. This is important. First exit the window manager (using a menu). After that you still have to exit Linux by typing `exit`.