Compiling a C program

It is recommended to first add a user (see separate tutorial) and compile a program being this user and not root. Very simple programs can be compiled using the following command, where *progname* is the name of the program.

```
cc progname.c -o progname
```

More elaborated programs use libraries. If that is the case, using make is much more convenient. In this little tutorial, we are going to compile a game using make.

Type cd /usr/scr/games to look at the source files of the text based games provided with 2.11 BSD. In this directory, there is also a Makefile, which could be used to compile all games provided, but there is nothing to learn from that. But you can inspect the Makefile to see what libraries are needed to compile any given game.

We are going to compile the game worms.

Go to your home directory and create a working directory there for this purpose. Go into that directory

```
cd
mkdir worms
cd worms
```

Copy the source of worms into your current working directory (.) and change permissions so that you can write to it.

```
cp /usr/src/games/worms.c .
chmod 744 worms.c
```

Now make a Makefile for the compilation. See the separate little tutorial for vi if you are not familiar with vi.

```
vi Makefile
```

Enter the following text, using <tab> to intend the cc line.

```
CFLAGS= -O
SEPFLAG= -i
worms: worms.c
     cc -o worms ${CFLAGS} ${SEPFLAG} worms.c -lcurses -ltermcap
```

Worms obviously uses the libraries curses and termcap. Once finished, you can use make to compile worms.c:

```
make
```

After it has been compiled successfully, you can execute it with

```
./worms
```

You can exit the little program using <ctrl>-c.

The synax of the C language used up to the early 1980ies such as the Berkley version used on 2.11 BSD is different from modern C versions. You can download the first edition of Dennis Ritchie and Brian Kernighans book “The C Programming language, First Edition”, available at https://archive.org/details/TheCProgrammingLanguageFirstEdition.