

Advanced Unix,
Basic C,
Program Compilation

man pages

- manual pages
 - man bash
 - man man
 - man ls

```
MAN(1)                                Manual pager utils                                MAN(1)

NAME
  man - an interface to the on-line reference manuals

SYNOPSIS
  man [-c|-w|-tZ] [-H[browser]] [-T[device]] [-X[dpi]] [-adhu7V] [-i|-I]
  [-m system[...]] [-L locale] [-p string] [-C file] [-M path] [-P
  pager] [-r prompt] [-S list] [-e extension] [--warnings [warnings]]
  [[section] page ...] ...
  man -l [-7] [-tZ] [-H[browser]] [-T[device]] [-X[dpi]] [-p string] [-P
  pager] [-r prompt] [--warnings[warnings]] file ...
  man -k [apropos options] regexp ...
  man -f [whatis options] page ...

DESCRIPTION
  man is the system's manual pager. Each page argument given to man is
  normally the name of a program, utility or function. The manual page
  associated with each of these arguments is then found and displayed. A
  section, if provided, will direct man to look only in that section of
  the manual. The default action is to search in all of the available
  sections, following a pre-defined order and to show only the first page
  found, even if page exists in several sections.
```

also available online:

google → man bash

(may be different than what is on your machine)

man pages - Library Function

FOPEN(3)

Name

fopen, fdopen, freopen - stream open functions

Synopsis

#include <stdio.h>

FILE *fopen(const char *path, const char *mode);

Description

(arguments or command line options are listed here)

Return Value

Errors

See Also

Referenced By

Simple C Program Editing

- Create a directory called CS300 in your Documents folder
- Change into the CS300 directory
- Open up a simple text editor called Geany in the Integrated Environment

```
shereen@ralph:~/Documents/CS300> geany &
```

- The & causes the program to be launched in the background so you can still use the command line.

Create the C program

Differences from C++ ?

```
/* this is a comment */
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    printf ("hello world\n");
```

```
    return 0;
```

```
}
```

Save, Build, Execute

- Save your program in Documents/CS300 with the name helloworld.c
- Change into the CS300 directory to see that the file helloworld.c exists
- Hit the Build button
 - only works with no configuration for simple projects
 - what shows up in the bottom window?
- Hit the Execute button

Running a Program from the Terminal

- List the contents of CS300 in the terminal now
 - What's changed from before?
- To run helloworld type `./helloworld`
- Note that in Linux, the executable does not have an extension such as (.exe) as it does in Windows

More UNIX Commands

Command / Symbol	Meaning
<code>tar czf file.tar.gz files...</code>	use the tar utility to compress file(s)
<code>tar xzf file.tar.gz</code>	use the tar utility to decompress file(s)
<code>./helloworld > outputfile</code>	save the executable results in outputfile
<code>./helloworld >> outputfile</code>	append the execution results to the end of outputfile
<code>./helloworld less</code>	pipe the output of helloworld to the input of less (useful if the output results are more than a screen in length)

Build on the command line

```
shereen@lisa:~> gcc -Wall -o runMe helloworld.c -g  
shereen@lisa:~> ./runMe
```

- `gcc`
- `-Wall`
- `-o`
- `runMe`
- `-g`
- What happens if you omit `runMe`?

Separate Compilation

```
~> gcc -Wall -c -o helloworld.o helloworld.c -g
```

```
~> gcc -Wall -o runMe helloworld.o -g
```

```
~> ls -altr
```

- `gcc`
 - `-c`
- `ls`
 - `-a`
 - `-l`
 - `-t`
 - `-r`

Problems

- tar up the file helloworld.c
- Copy (not move) the tarred file to the parent directory
- Change to the parent directory and untar the file
- Compile the untarred file
- Run the executable
- Capture the execution results in a file called **rslts**
- Type the command less **rslts**

C Topics

```
/* this is a comment */  
  
#include <stdio.h>  
  
int main ()  
{  
    int value;  
  
    scanf ("%d", &value);  
  
    printf ("hello world %d\n", value);  
    return 0;  
}
```