Hosted C and Compiling C code with Linux

Adam Boult (www.bou.lt)

November 2, 2023

Contents

Preface

This is a live document, and is full of gaps, mistakes, typos etc.

Part I

GNU binutils

Chapter 1

GNU Binutils, including as (assembler), ld (linker), gprof (profiler), objdump

1.1 Introduction

1.1.1 Introduction

Part II

Compiling

Chapter 2

GNU Compiler Collection (gcc) and the GNU C Compiler (also gcc)

2.1 Introduction

2.1.1 Introduction

compile to object,

compile to assembly, links and runtime libraries? or is in make?

objdump; returning stderr and stdout. raise

Chapter 3

C headers for linux

3.1 Introduction

3.1.1 Introduction

3.1.2 getopt

```
#include <unistd.h>

int getopt(int argc, char *const argv[],
const char *optstring);

extern char *optarg;
extern int optind, opterr, optopt;
```

allows

```
myScript -a s
```

3.1.3 Longer

```
#include <getopt.h>

int getopt_long(int argc, char *const argv[],
           const char *optstring,
           const struct option *longopts, int *longindex);
int getopt_long_only(int argc, char *const argv[],
           const char *optstring,
           const struct option *longopts, int *longindex);
```

allows

```
myScript --a something
```

Chapter 4

GNU Debugger (gdb)

4.1 Introduction

4.1.1 Introduction

debugger. can step through, stop on specific functions etc

Chapter 5

make

5.1 Introduction

5.1.1 Introduction

Makefile; target:, .PHONY clean; clean: (means that doesn't look to see if clean exists because declared as phony.)

Part III

Hosted C: C standard library: stdlib.h

Chapter 6

stdlib.h: Maths (abs, labs, div_t, ldiv_t)

6.1 Introduction

6.1.1 Introduction

int abs(int x)

long int labs(long x);

div_t div(int numer, int denom); ldiv_t ldiv(long int numer, long int denom);

div_t and ldiv_t types are introduced in stdlib.h

Chapter 7

stdlib.h: abort() and exit()

7.1 Introduction

7.1.1 Introduction

void abort(void); void exit(int status);

Chapter 8

stdlib.h: Type conversions (atof, atol, strtod, strtol, strtoul)

8.1 Introduction

8.1.1 Introduction

character stuff with stdlib.h double atof int atoi long int atol long strtod long int strtol usigned long int strtoul

Chapter 9

stdlib.h: The heap

9.1 Introduction

9.1.1 Introduction

variable length array using the heap or the stack

```
void my_function(int n){
    float vals[n];
    return process(n, vals);
}
```

stack approach above. deleted when function ends

```
void my_function(int n){
    [something using malloc]
}
```

heap memory: allocated at run time needs to be manually removed

stack overflow + stack hits point reserved for heap?

+ memory leaks

space for stack limitated with need to reserve space to prevent overlapping with heap heap in c doesn't mean data structure. heap as in pile, or pool, just a description. just synonym for pile, area etc. Heap in c. Allocate with malloc?

9.1.2 malloc

malloc (memory allocation):

```
int * ptr = (int *) malloc(10 * sizeof(int));
```

9.1.3 calloc

calloc (contiguous allocation):

```
int * ptr = (int *) calloc(10, sizeof(int));
```

calloc initiates all to 0

9.1.4 free

can free

```
int * ptr = (int *) malloc(10 * sizeof(int));
free(ptr);
```

or

```
int * ptr = (int *) calloc(10, sizeof(int));
free(ptr);
```

9.1.5 realloc

```
int * ptr = (int *) malloc(10 * sizeof(int));
ptr = realloc(ptr, 20*sizeof(int));
```

Chapter 10

stdlib.h: Algorithms (qsort, bsearch, rand)

10.1 Introduction

10.1.1 Introduction

10.1.2 qsort()

sort in standard library (qsort)

10.1.3 RAND

```
void srand(unsigned int seed);
int rand(void); (returns between 0 and RAND_MAX)
```

10.1.4 bsearch

```
void *bsearch()
```

Part IV

Hosted C: C standard library

Chapter 11

math.h, complex.h and tgmath.h

11.1 Introduction

11.1.1 math.h

11.1.2 complex.h

Complex numbers.

11.1.3 tgmath.h

Type-generic math.

Chapter 12

fenv.h

12.1 Introduction

12.1.1 Introduction

Chapter 13

setjmp.h

13.1 Introduction

13.1.1 Introduction

Chapter 14

Getting and setting locale using locale.h (eg keyboard, date format)

14.1 Introduction

14.1.1 Introduction

Chapter 15

Time and date with time.h

15.1 Introduction

15.1.1 Introduction

Date or time handling.

Chapter 16

Working with strings with string.h and ctype.h

16.1 Introduction

16.1.1 Introduction

Chapter 17

Specified width integers with inttypes.h (replacing freestanding C's stdint.h)

17.1 Introduction

17.1.1 Introduction

Chapter 18

Errors and signals with errno.h, signal.h and assert.h

18.1 Introduction

18.1.1 Introduction

Chapter 19

Wide characters including Unicode with uchar.h, wchar.h and wctype.h

19.1 Introduction

19.1.1 Introduction

Chapter 20

stdio.h

20.1 Introduction

20.1.1 Introduction

20.1.2 stdio

stdio allows arguments but not as flags. allows

```
myScript something
```

Chapter 21

Multi-threading with threads.h and stdatomic.h

21.1 Introduction

21.1.1 Introduction

Chapter 22

ctype.h

22.1 Introduction

22.1.1 Introduction

Part V

Analysing programs

Chapter 23

Static analysis

23.1 Introduction

23.1.1 Introduction

frama-C?, formal verification

Chapter 24

Dynamic analysis

24.1 Introduction

24.1.1 Introduction

unit tests (inc asserts) and integration tests