

1^η Αναφορά στα Λειτουργικά Συστήματα

Ιωάννης Αλεξόπουλος (03117001)

Αλέξανδρος Κυριακάκης (03112163)

June 6, 2020

1 Σειρά

1.1 Άσκηση

1.1.1 Header

Η επικεφαλίδα είναι απαραίτητη για τους εξής λόγους:

- Την δήλωση των συναρτήσεων που θα χρησιμοποιηθούν και των ορισμάτων τους, για να ξέρει ο compiler τι θα χρειαστεί και που θα το βρει.
- Για optimized compiling αφού κάποιες συναρτήσεις μπορεί να χρησιμοποιηθούν σε πάνω από ένα αρχεία.

1.1.2 Makefile

```
1 all: zing zing2 zing2.o main.o
2 zing: zing.o main.o
3   gcc -o zing zing.o main.o
4
5 zing2: zing2.o main.o
6   gcc -o zing2 zing2.o main.o
7
8 zing2.o: zing2.c
9   gcc -Wall -c zing2.c
10
11 main.o: main.c
12   gcc -Wall -c main.c
```

1.1.3 zing2.c

```
1 #include <stdio.h>
2 #include <unistd.h>
3 void zing(void){
4   printf("Hello Giannis and Alexandros, %s\n",getlogin());
5 }
```

Output zing2.c

```
1 Hello Giannis and Alexandros, oslab18
```

1.1.4 500 Συναρτήσεις

Για την αποφυγή του αυξημένου Compilation Time, συνίσταται ο διαχωρισμός των συναρτήσεων σε διαφορετικά αρχεία "*.c", αν όχι η κάθε μία σε διαφορετικό, τότε π.χ. ανα πέντε. Έτσι, για κάθε ένα αρχείο ".c" θα κατασκευάσουμε ένα αρχείο object ".o" το οποίο θα περιέχεται στο Makefile μαζί με τα αντίστοιχα αρχεία που θα χρησιμοποιούν την/τις συναρτήσεις που περιέχει. Άρα για μία αλλαγή σε μία μόνο συνάρτηση θα γίνει εκ νέου compilation μόνο στα αρχεία που τροποποιήθηκαν δηλαδή μόνο σε αυτό του περιέχει την συνάρτηση που αλλάχθηκε.

1.1.5 Name Problem

Η εντολή που έτρεξε ο συνάδελφος μου, "gcc -Wall -o foo.c foo.c" σημαίνει πάρε το argument ακριβώς μετά το "-o" και δημιούργησε ένα εκτελέσιμο με αυτό το όνομα "foo.c" μεταγλωττίζοντας όποιο άλλο όρισμα έχει δωθεί "foo.c". Έτσι μετά την κλήση αυτή θα υπάρχει στον τρέχων κατάλογο ένα εκτελέσιμο αρχείο με όνομα "foo.c" που έχει παραχθεί από το "foo.c". Έτσι θα γίνει overwrite το όνομα αρχείου foo.c και θα χαθεί με αυτόν τον τρόπο το αρχείο με τον κώδικα.

1.2 Άσκηση

1.2.1 fconc.c

```
1  #include <sys/types.h>
2  #include <sys/stat.h>
3  #include <fcntl.h>
4  #include <stdio.h>
5  #include <stdlib.h>
6  #include <unistd.h>
7  #include <string.h>
8  int doWrite(int fd, const char *buff, int len){
9      size_t idx = 0;
10     ssize_t wcnt = 0;
11     do {
12         wcnt = write(fd, buff + idx, len - idx);
13         if (wcnt == -1) {
14             perror("write");
15             return 1;
16         }
17         idx += wcnt;
18     } while (idx < len);
19 }
20
21 int write_file(int fd, const char *infile){
22     char buff[1024];
23     ssize_t rcnt, fd_open;
24     fd_open = open(infile, O_RDONLY);
25     if (fd_open == -1 ){
```

```

26         perror("open");
27         exit(1);
28     }
29     //printf("opened file: %s\n", infile);
30     for (;;) {
31         rcnt = read(fd_open, buff, sizeof(buff)-1);
32         if (rcnt == 0) /* end-of-file */
33             return 0;
34         if (rcnt == -1) {
35             perror("read");
36             return 1;
37         }
38         int len = strlen(buff);
39         doWrite(fd, buff, len);
40     }
41     close(fd_open);
42
43 }
44 int main(int argc, char** argv) {
45     if (argc < 3 || argc > 4) {
46         printf("Usage: ./fconc infile1 infile2 [outfile (default:fonc.out)]\n");
47         // printf("argc = %d\n", argc);
48     }
49     else {
50         int fd_out, oflags, mode;
51         oflags = O_CREAT | O_WRONLY | O_TRUNC;
52         mode = S_IRUSR | S_IWUSR;
53         if (argc == 3) {
54             fd_out = open("fconc.out", oflags, mode);
55         }
56         else {
57             fd_out = open(argv[3], oflags, mode);
58         }
59         if (fd_out == -1) {
60             perror("open");
61             exit(1);
62         } else {
63
64             // printf("file opened\n");
65             write_file(fd_out, argv[1]);
66             write_file(fd_out, argv[2]);
67             close(fd_out);
68         }
69     }
70
71     return 0;
72 }

```

Output fconc.c

```
1 Goodbye,  
2 And thanks for all the fish!  
3 -
```

Strace fconc.c Output

```
1 execve("./fconc", ["/fconc", "A", "B"], [/* 21 vars */]) = 0  
2 brk(0) = 0x701000  
3 access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)  
4 mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0  
   x7fee4d619000  
5 access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)  
6 open("/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3  
7 fstat(3, {st_mode=S_IFREG|0644, st_size=29766, ...}) = 0  
8 mmap(NULL, 29766, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fee4d611000  
9 close(3) = 0  
10 access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)  
11 open("/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3  
12 read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0P\34\2\0\0\0\0"... , 832) =  
   832  
13 fstat(3, {st_mode=S_IFREG|0755, st_size=1738176, ...}) = 0  
14 mmap(NULL, 3844640, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0  
   x7fee4d050000  
15 mprotect(0x7fee4d1f1000, 2097152, PROT_NONE) = 0  
16 mmap(0x7fee4d3f1000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE  
   , 3, 0x1a1000) = 0x7fee4d3f1000  
17 mmap(0x7fee4d3f7000, 14880, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS  
   , -1, 0) = 0x7fee4d3f7000  
18 close(3) = 0  
19 mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0  
   x7fee4d610000  
20 mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0  
   x7fee4d60f000  
21 mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0  
   x7fee4d60e000  
22 arch_prctl(ARCH_SET_FS, 0x7fee4d60f700) = 0  
23 mprotect(0x7fee4d3f1000, 16384, PROT_READ) = 0  
24 mprotect(0x7fee4d61b000, 4096, PROT_READ) = 0  
25 munmap(0x7fee4d611000, 29766) = 0  
26 open("fconc.out", O_WRONLY|O_CREAT|O_TRUNC, 0600) = 3  
27 open("A", O_RDONLY) = 4  
28 read(4, "Goodbye,\n", 1023) = 9  
29 write(3, "Goodbye,\n\360\226|", 12) = 12  
30 read(4, "", 1023) = 0  
31 open("B", O_RDONLY) = 5  
32 read(5, "and thanks for all the fish!\n", 1023) = 29  
33 write(3, "and thanks for all the fish!\n", 29) = 29  
34 read(5, "", 1023) = 0  
35 close(3) = 0  
36 exit_group(0) = ?  
37 +++ exited with 0 +++
```