

Αναφορά 1^{ης} σειράς ασκήσεων στα
Λειτουργικά Συστήματα

ΟΜΑΔΑ: oslaba30

Αικατερίνη Σταύρου

Όλγα Γιαννούχου

ΑΣΚΗΣΗ 1.1

ΑΠΑΝΤΗΣΕΙΣ ΣΤΙΣ ΕΡΩΤΗΣΕΙΣ

Ερώτηση 1

Η επικεφαλίδα χρησιμοποιείται για την καλύτερη διαχείριση/μεταγλώττιση του κώδικα. Με την επικεφαλίδα, μπορούμε να καλέσουμε συναρτήσεις, τις οποίες έχουμε δημιουργήσει σε διαφορετικά αρχεία, σε διάφορα προγράμματα χωρίς να χρειαστεί να τις δηλώσουμε ξανά. Ουσιαστικά, με την επικεφαλίδα μπορεί ο preprocessor να εντοπίσει το αρχείο όπου δηλώνεται η συνάρτηση και ο compiler να μπορέσει να μεταγλωττίσει το πρόγραμμα, συνδέοντας τα κατάλληλα αρχεία μεταξύ τους.

Ερώτηση 2:

Το Makefile που χρησιμοποιήθηκε για τη μεταγλώττιση και σύνδεση των αρχείων:

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

Ερώτηση 3:

Για τα προγράμματα zing, zing2 η main.c είναι η εξής:

```
1  #include "zing.h"
2
3  int main (int argc, char **argv) {
4      zing();
5      return 0;
6
7  }
```

Το header file της συνάρτησης zing

```
1 void zing(void);
```

Η συνάρτηση zing() αλλαγμένη για το πρόγραμμα zing2

```
1  #include <unistd.h>
2  #include <stdio.h>
3
4  void zing(void) {
5      printf("Hello this is Katerina and Olga from %s \n",
6      getlogin());
7  }
```

Έξοδος προγράμματος zing

Hello, oslaba30

Έξοδος προγράμματος zing

Hello this is Katerina and Olga from oslaba30

Ερώτηση 4:

Για να αντιμετωπίσουμε το πρόβλημα που προκύπτει στο αρχείο με τις 500 συναρτήσεις θα ήταν χρήσιμο να μοιράσουμε αυτές τις συναρτήσεις σε περισσότερα αρχεία. Επίσης με τη χρήση του Makefile μπορούμε να κάνουμε αλλαγές σε ένα μέρος του κώδικα(π.χ. σε μία από τις 500 συναρτήσεις) και να ενημερώνεται μόνο το κατάλληλο object file, οδηγώντας τελικά στη μείωση του χρόνου μεταγλώττισης.

Ερώτηση 5:

Με την εντολή : `gcc -Wall -o foo.c foo.c`

Το output file ταυτίζεται με το ίδιο αρχείο foo.c το οποίο μεταγλωττίζουμε. Αυτό έχει ως αποτέλεσμα να γίνει overwrite και το source file(foo.c) μας να μην υπάρχει πια.

ΑΣΚΗΣΗ 1.2

(doWrite.c)

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

(write_file.c)

```
1 #include <unistd.h>
2 #include <stdio.h>
3 #include <stdlib.h>
4 #include <sys/types.h>
5 #include <sys/stat.h>
6 #include <fcntl.h>
7 #include <string.h>
8
9 int write_file(int fd, const char *infile){
10
11     char buff[1024];
12     ssize_t rcnt, fd1;
13     fd1 = open(infile, O_RDONLY);
14
15     if(fd1 == -1) {
16         perror("open");
17         exit(1);
18     }
19
20     for(;;){
21         rcnt = read(fd1, buff, sizeof(buff)-1);
22
23         if(rcnt == 0)
24             return 0;
25
26         if(rcnt == -1){
27             perror("read");
28             return 1;
29         }
30
31         buff[rcnt] = '\0';
32
33         int len;
34         len = strlen(buff);
35         doWrite(fd, buff, len);
36     }
37 close(fd1);
38 }
```

(main.c)

```

1 #include <unistd.h>
2 #include <stdio.h>
3 #include <stdlib.h>
4 #include <sys/types.h>
5 #include <sys/stat.h>
6 #include <fcntl.h>
7 #include "headerf.h"
8
9 int main(int argc, char **argv) {
10
11     if(argc < 3 || argc > 4) {
12
13         printf(" %s \n ", "Usage: ./fconc infile1 infile2
14 [outfile (default:fconc.out)]");
15
16     }
17
18     else {
19
20         int fd_conc, oflags, mode;
21         oflags = O_CREAT | O_WRONLY | O_TRUNC;
22         mode = S_IRUSR | S_IWUSR;
23         if(argc == 3) {
24
25             fd_conc = open("fconc.out", oflags, mode);
26
27         }
28
29         else {
30
31             fd_conc = open(argv[3], oflags, mode);
32
33         }
34
35         if(fd_conc == -1) {
36
37             perror("open");
38             exit(1);
39
40         } else {
41
42             write_file(fd_conc, argv[1]);
43             write_file(fd_conc, argv[2]);
44             close(fd_conc);
45
46         }
47
48     }
49
50     return 0;
51
52 }

```


Αποτέλεσμα **strace cat ./fconc A B C**

\$ strace ./fconc A B C

```
execve("./fconc", ["/fconc", "A", "B", "C"], [/ * 21 vars */]) = 0
brk(0) = 0xcb9000
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or
directory)
mmap(NULL, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f88ae6b0000
access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or
directory)
open("/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=29766, ...}) = 0
mmap(NULL, 29766, PROT_READ, MAP_PRIVATE, 3, 0) =
0x7f88ae6a8000
close(3) = 0
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or
directory)
open("/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
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
fstat(3, {st_mode=S_IFREG|0755, st_size=1738176, ...}) = 0
mmap(NULL, 3844640, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f88ae0e7000
mprotect(0x7f88ae288000, 2097152, PROT_NONE) = 0
mmap(0x7f88ae488000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1a1000) =
0x7f88ae488000
mmap(0x7f88ae48e000, 14880, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) =
0x7f88ae48e000
close(3) = 0
mmap(NULL, 4096, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f88ae6a7000
mmap(NULL, 4096, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f88ae6a6000
mmap(NULL, 4096, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f88ae6a5000
arch_prctl(ARCH_SET_FS, 0x7f88ae6a6700) = 0
mprotect(0x7f88ae488000, 16384, PROT_READ) = 0
mprotect(0x7f88ae6b2000, 4096, PROT_READ) = 0
munmap(0x7f88ae6a8000, 29766) = 0
open("C", O_WRONLY|O_CREAT|O_TRUNC, 0600) = 3
open("A", O_RDONLY) = 4
read(4, "Goodbye, \n", 1023) = 10
```

```
write(3, "Goodbye, \n", 10)      = 10
read(4, "", 1023)                = 0
open("B", O_RDONLY)              = 5
read(5, "and thanks for all the fish!\n", 1023) = 29
write(3, "and thanks for all the fish!\n", 29) = 29
read(5, "", 1023)                = 0
close(3)                         = 0
exit_group(0)                    = ?
+++ exited with 0 +++
```