

Make Tutorial

Μια εισαγωγή στο εργαλείο make

Πρώτη επαφή με το make

Εγκατάσταση λογισμικού

```
# ./configure
```

```
# make
```

```
# make install
```

Δημιουργία νέου πυρήνα

```
# make menuconfig
```

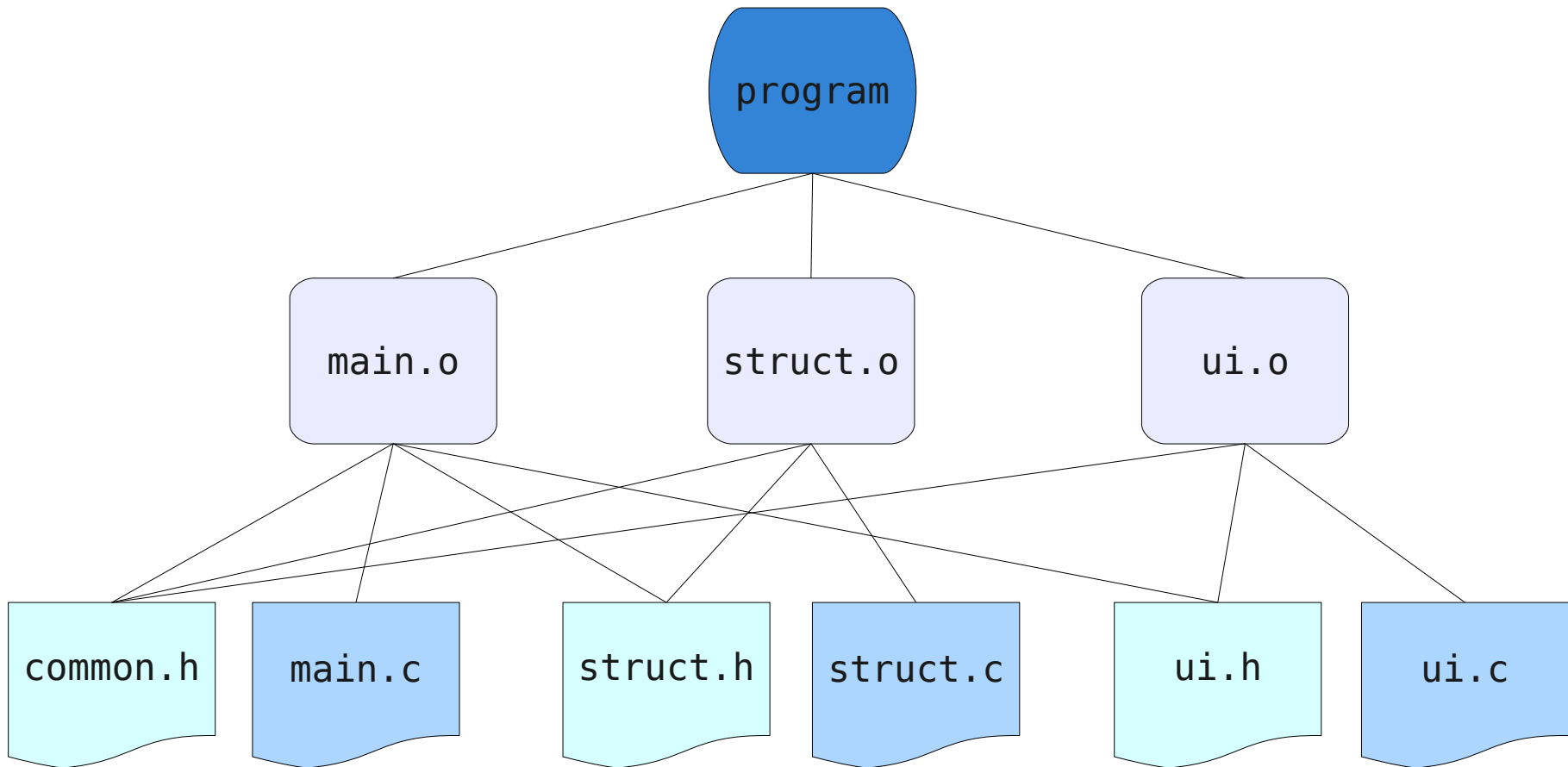
```
# make
```

```
# make modules_install
```

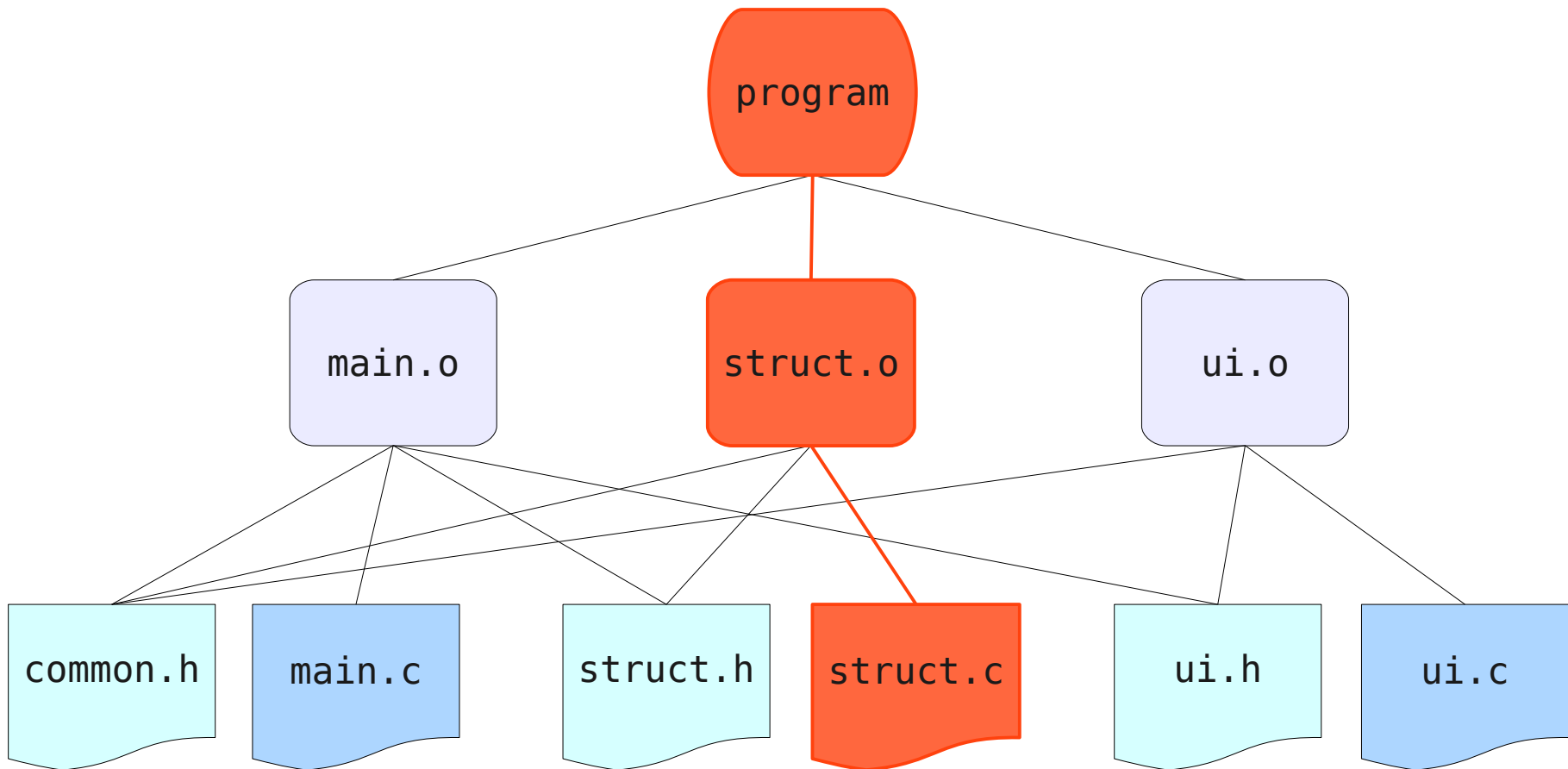
Τι κάνει το make



Πώς δουλεύει το make (1)



Πώς δουλεύει το make (2)



Δομή ενός makefile

```
target: [component ...]  
    <TAB> [command 1]  
        .  
        .  
        .  
    <TAB> [command n]
```

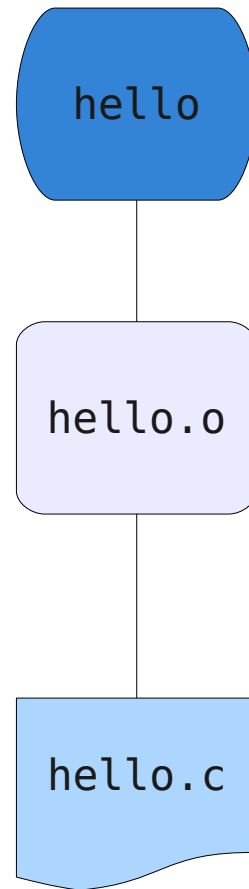
Σχόλια

Αυτή η γραμμή είναι ένα σχόλιο.

Κλήση της εντολής make

```
$ make [target]
```

Το πρώτο σας makefile (1)



Το πρώτο σας makefile (2)

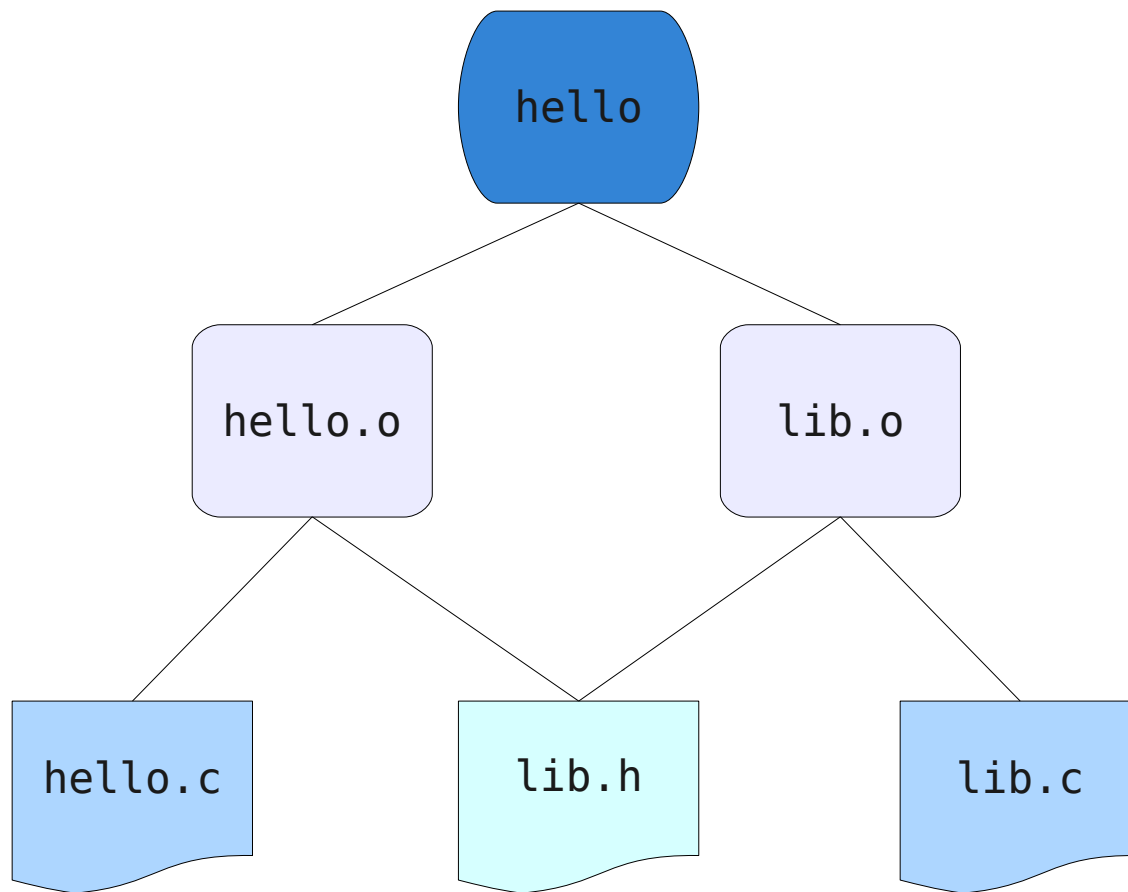
Makefile

```
hello: hello.c  
    gcc -o hello hello.c
```

Το πρώτο σας makefile (3)

```
$ make
gcc -o hello hello.c
$ ls
Makefile hello hello.c
$ make
make: `hello' is up to date.
$ touch hello.c
$ make
gcc -o hello hello.c
```

Πρόγραμμα με βιβλιοθήκη (1)



Πρόγραμμα με βιβλιοθήκη (2)

Makefile

```
hello: hello.c lib.c lib.h  
      gcc -o hello hello.c lib.c
```

Πρόγραμμα με βιβλιοθήκη (3)

```
$ make
```

```
gcc -o hello hello.c lib.c
```

```
$ ls
```

```
Makefile hello hello.c lib.c lib.h
```

```
$ touch lib.c
```

```
$ make
```

```
gcc -o hello hello.c lib.c
```

Πρόγραμμα με βιβλιοθήκη (4)

Makefile

```
# Το πρώτο target είναι το default.  
hello: hello.o lib.o  
    gcc -o hello hello.o lib.o  
  
hello.o: hello.c lib.h  
    gcc -c -o hello.o hello.c  
lib.o: lib.c lib.h  
    gcc -c -o lib.o lib.c
```

Πρόγραμμα με βιβλιοθήκη (5)

```
$ make
```

```
gcc -c -o hello.o hello.c
```

```
gcc -c -o lib.o lib.c
```

```
gcc -o hello hello.o lib.o
```

```
$ ls
```

```
Makefile  hello.o  lib.c  lib.o
```

```
hello    hello.c  lib.h
```

```
$ touch lib.c
```

```
$ make
```

```
gcc -c -o lib.o lib.c
```

```
gcc -o hello hello.o lib.o
```

Μεταβλητές

Εντολή εκχώρησης
`VARIABLE = value`

Προσπέλαση περιεχομένου
`$(VARIABLE)`

Πρόγραμμα με βιβλιοθήκη (6)

Makefile

```
# Υποστήριξη για GDB.  
DEBUG = -g  
  
# Το πρώτο target είναι το default.  
hello: hello.o lib.o  
    gcc -o hello hello.o lib.o  
  
hello.o: hello.c lib.h  
    gcc $(DEBUG) -c -o hello.o hello.c  
lib.o: lib.c lib.h  
    gcc $(DEBUG) -c -o lib.o lib.c
```

Ειδικές μεταβλητές

<code>\$@</code>	όνομα του target
<code>\$^</code>	το πρώτο component
<code>\$*</code>	όλα τα components

Πρόγραμμα με βιβλιοθήκη (7)

Makefile

```
# Υποστήριξη για GDB.  
DEBUG = -ggdb  
  
# Το πρώτο target είναι το default.  
hello: hello.o lib.o  
    gcc -o $@ $^  
  
hello.o: hello.c lib.h  
    gcc $(DEBUG) -c -o $@ $<  
lib.o: lib.c lib.h  
    gcc $(DEBUG) -c -o $@ $<
```

Αυτόματες μεταβλητές

Μπορείτε να τις δείτε με την εντολή

```
$ make -p
```

Πρόγραμμα με βιβλιοθήκη (8)

Makefile

```
# Υποστήριξη για GDB.  
DEBUG = -ggdb  
  
# Το πρώτο target είναι το default.  
hello: hello.o lib.o  
    $(CC) -o $@ $^  
  
hello.o: hello.c lib.h  
    $(CC) $(DEBUG) -c -o $@ $<  
lib.o: lib.c lib.h  
    $(CC) $(DEBUG) -c -o $@ $<
```

Αυτόματα rules

YES PLEASE!

Πρόγραμμα με βιβλιοθήκη (9)

Makefile

```
# Υποστήριξη για GDB.  
CFLAGS = -ggdb  
  
# Το πρώτο target είναι το default.  
(1) hello: hello.o lib.o  
  
(2) hello.o: hello.c lib.h  
  
(2) lib.o: lib.c lib.h
```

```
(1) $(CC) $(LDFLAGS) $(TARGET_ARCH) $^ $(LOADLIBES) $(LDLIBS) -o $@  
(2) $(CC) $(CFLAGS) $(CPPFLAGS) $(TARGET_ARCH) -c -o $@ $<
```

Πρόγραμμα με βιβλιοθήκη (10)

```
$ make
```

```
cc -ggdb -c -o hello.o hello.c
```

```
cc -ggdb -c -o lib.o lib.c
```

```
cc hello.o lib.o -o hello
```

Ένας ακόμη τελεστής εκχώρησης

`VARIABLE += value`

Πρόγραμμα με βιβλιοθήκη (11)

Makefile

```
# Υποστήριξη για GDB.  
CFLAGS += -ggdb  
# Προειδοποιήσεις κατά τη μεταγλώττιση.  
CFLAGS += -Wall -Wextra  
  
# Το πρώτο target είναι το default.  
hello: hello.o lib.o  
  
hello.o: hello.c lib.h  
  
lib.o: lib.c lib.h
```

Απόκρυψη εκτελούμενης εντολής

Χρήση του χαρακτήρα @

```
target: [component ...]  
        <TAB> @[command]
```

Συχνά targets

<code>all</code>	το default target
<code>clean</code>	διαγραφή των αρχείων του <code>all</code>

Συχνά targets: all

Makefile

```
all: targets
```

```
# ...
```

Συχνά targets: clean

Makefile

```
# ...
```

```
clean:  
    $(RM) files
```

Η ντιρεκτίβα .PHONY (1)

```
$ touch clean
```

```
$ make clean
```

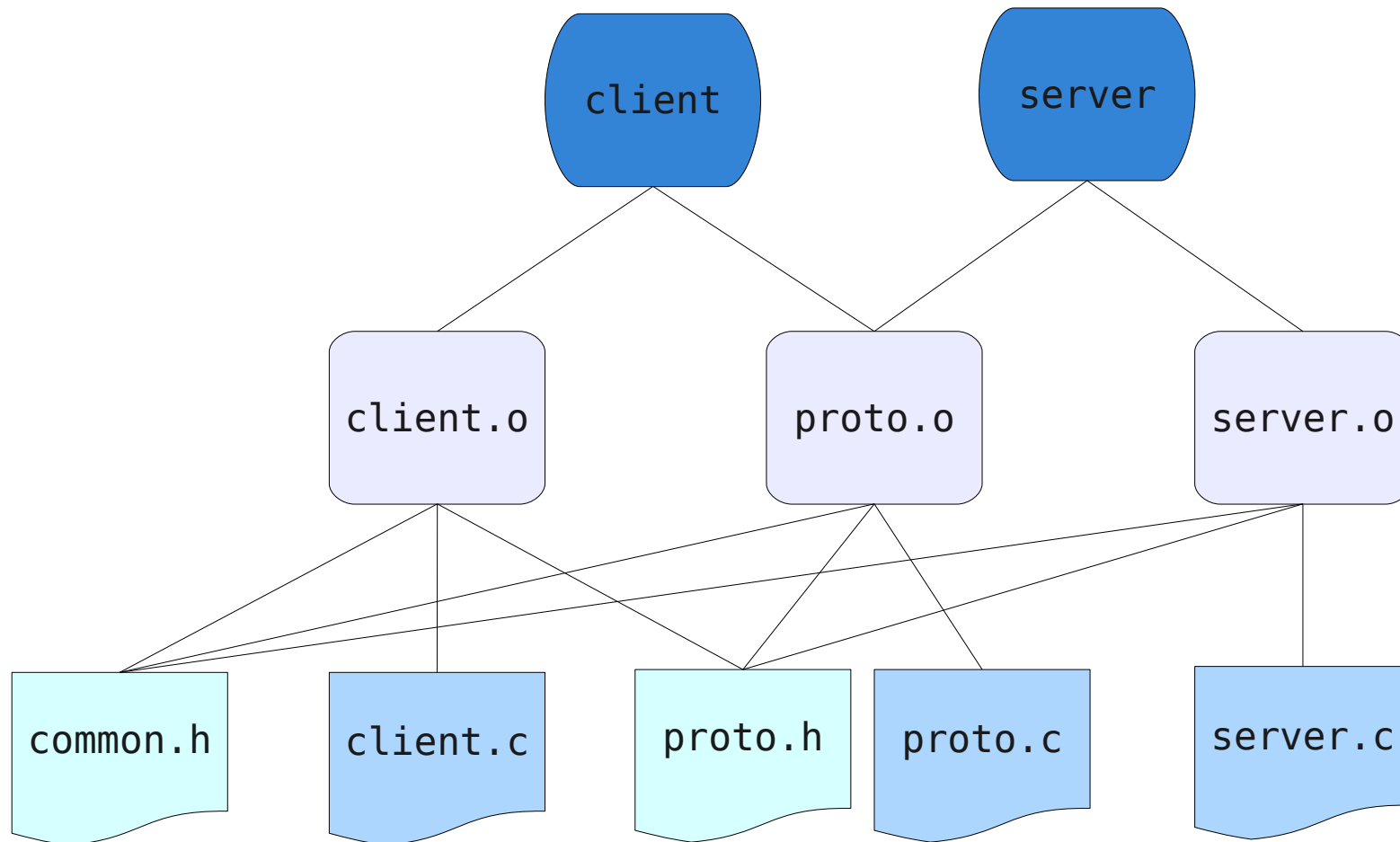
```
make: `clean' is up to date.
```

Η ντιρεκτίβα .PHONY (2)

Makefile

```
# ...  
  
.PHONY: clean  
clean:  
    $(RM) files
```

Πρόγραμμα client server (1)



Πρόγραμμα client server (2)

Makefile

```
all: client server

TARGETS = client server
client: client.o proto.o
server: server.o proto.o

OBJECTS = client.o proto.o server.o
client.o: client.c common.h proto.h
proto.o: proto.c common.h proto.h
server.o: server.c common.h proto.h

.PHONY: clean
clean:
    $(RM) $(TARGETS) $(OBJECTS)
```

Πρόγραμμα client server (3)

```
$ make
```

```
cc      -c -o client.o client.c
```

```
cc      -c -o proto.o proto.c
```

```
cc      client.o proto.o      -o client
```

```
cc      -c -o server.o server.c
```

```
cc      server.o proto.o      -o server
```

```
$ ls
```

```
Makefile  client.o  proto.h  server.c  
client   common.h  proto.o  server.o  
client.c  proto.c  server
```

Πρόγραμμα client server (4)

```
$ touch client.c
$ make
cc      -c -o client.o client.c
cc      client.o proto.o      -o client
$ make clean
rm -f client server client.o proto.o
server.o
$ ls
Makefile      common.h      proto.h
client.c      proto.c       server.c
```

Ερωτήσεις;

Παράρτημα

Σπάσιμο κώδικα σε πολλά αρχεία

Αρχικός κώδικας

hello.c

```
#include <stdio.h>

int main(int argc, char *argv[])
{
    // This is a VERY useful line.
    printf("hello, world\n");

    return 0;
}
```

Δημιουργία νέας συνάρτησης

hello.c

```
#include <stdio.h>

int main(int argc, char *argv[])
{
    // This is a VERY useful line.
    printf("hello, world\n");

    return 0;
}
```

```
#include <stdio.h>

void do_something_useful(void);

int main(int argc, char *argv[])
{
    // Call the VERY useful function.
    do_something_useful();

    return 0;
}

void do_something_useful(void)
{
    printf("hello, world\n");
}
```

Μεταφορά σε νέο αρχείο (1)

hello.c

```
#include <stdio.h>

void do_something_useful(void);

int main(int argc, char *argv[])
{
    // Call the VERY useful function.
    do_something_useful();

    return 0;
}

void do_something_useful(void)
{
    printf("hello, world\n");
}
```

useful.h

```
void do_something_useful(void);
```

useful.c

```
#include <stdio.h>

void do_something_useful(void)
{
    printf("hello, world\n");
}
```

Μεταφορά σε νέο αρχείο (2)

hello.c

```
#include "useful.h"

int main(int argc, char *argv[])
{
    // Call the VERY useful function.
    do_something_useful();

    return 0;
}
```

useful.h

```
void do_something_useful(void);
```

useful.c

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

void do_something_useful(void)
{
    printf("hello, world\n");
}
```

Μεταγλώττιση και σύνδεση (1)

```
$ gcc hello.c useful.c # a.out
```

Μεταγλώττιση και σύνδεση (2)

```
$ gcc -c hello.c # hello.o  
$ gcc -c useful.c # useful.o  
$ gcc hello.o useful.o # a.out
```