

Liburutegiak

- Edozein programa egiten dugunean:Adb:

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

```
#include <stdlib.h>
```

```
int main(){
```

```
printf("kaixo");
```

```
exit(0);
```

```
}
```

Liburutegiak

- C lengoaian liburutegiak erabiltzen dira:

```
gcc programa.c -o programa
```

```
ldd programa //muestra las dependencias de  
bibliotecas compartidas
```

```
linux-vdso.so.1 => (0x00007fff01dfe000)
```

```
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6  
(0x00007f80dd5af000)
```

```
/lib64/ld-linux-x86-64.so.2 (0x00007f80dd990000)
```

- Zein paketetan datorren ikusteko:

```
dpkg -S /lib/x86_64-linux-gnu/libc.so.6
```

```
libc6:amd64: /lib/x86_64-linux-gnu/libc.so.6
```

Liburutegiak

- Paketearen informazioa ikusteko:

```
$ dpkg -l libc6:amd64
```

```
Deseado=desconocido(U)/Instalar/eliminaR/Purgar/retener(H)
```

```
| Estado=No/Inst/ficheros-Conf/desempaquetado/medio-conf/medio-inst(H)/espera-  
disparo(W)/pendiente-disparo
```

```
|/ Err?=(ninguno)/requiere-Reinst (Estado,Err: mayúsc.=malo)
```

```
||/ Nombre                Versión          Arquitectura    Descripción
```

```
+++-----  
=====
```

```
ii libc6:amd64            2.23-0ubuntu10 amd64            GNU C Library: Shared  
libraries
```

libc.Liburutegiak

- Sistemako programa askok erabiltzen dituzten **libc** paketeko funtzioak.
- Pakete honetan daude liburutegi partekatu garrantzitsuenak, hau da, Cko liburutegi estandarra eta liburutegi estandar matematikoa. Bi liburutegi hauek barik Linux sistemak ez du funtzionatu.

Liburutegiak

- Cko liburutegi estandarrak erabiliko dituen programa bat garatu nahi badugu, libc6-dev instalatu beharko dugu.

```
$sudo aptitude install libc6-dev
```

- Libc6-dev paketearen edukia:GNU C Library:

Development Libraries and Header Files

```
$sudo dpkg -L libc6-dev
```

Liburutegiak

- Cko liburutegi estandarrak erabiliko dituen programa bat garatzeko, /usr/include-n instalatuko diren goiburu fitxategiez gain (Adb:/usr/include/stdio.h), exekutagarria sortzeko beharko dituen liburutegiak ere instalatu behar dira(Adb:/usr/lib/x86-64-linux-gnu/libc.so edo libc.a)

```
sudo dpkg -S /usr/include/stdio.h
```

```
libc6-dev: /usr/include/stdio.h
```

Liburutegiak

- C-n idatzitako pogramek liburutegiak(.so eta .a) eta goiburu fitxategiak(.h) erabiltzen dituzte.
- C liburutegiko goiburu artxiboetan:
 - Liburutegietan barneratuta dauden funtzioen erazagupenak edo goiburuak ageri dira.
 - motak berriak eta makroak daude.
 - Aurre-prozesatzailea direktiba hauek prozesatzeaz arduratzen da: #include, #define, ...
 - Goiburuak /usr/include-n daude.

Liburutegia

- Liburutegiko artxiboak:
 - Estatikoak: Liburutegiko kodea exekutagarriaren barnean sartzen da.
 - Dinamikoak: Liburutegiko kodea exekuzio denboran zehar kargatzen da.

Liburutegiak

- Liburutegi dinamikoen abantailak estatikoekin parekatuz:
 - Exekutagarriek memoria gutxiago behar dute. Nahiz eta hainbat programetarik deitu, behin bakarrik kargatzen baitira.
 - Liburutegian aldaketak egiterakoan, ez da aplikazioa berriz konpilatu behar.

Liburutegiak antolatzea

- Sartu funtzioen kodeak .c luzapeneko iturri fitxategi batean.
- Sartu funtzioen erezagupenak edo goiburuak eta erabiliko diren mota berriak (typedefs, structs...) izen berdineko .h goiburu fitxategi batean.
- Bihurtu .c → .o, eta sartu .o-ak liburutegi batean.
- Liburutegiak /lib eta /usr/lib -en daude.
(<http://www.pathname.com/fhs/pub/fhs-2.3.html>)
/usr/lib : Libraries for programming and packages
 - Adb:/usr/lib/x86_64-linux-gnu/libc.so (garapen)/lib : Essential shared libraries and kernel modules
 - Adb:\$ldd programa.exe
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (erabilpen)

Liburutegiak antolatzea

```
modulu1.h:  
#ifndef _LIBURUTEGIA_1_H  
#define _LIBURUTEGIA_1_H  
int gehiketa (int a, int b);  
int kenketa (int a, int b);  
#endif
```

```
modulu1.c:  
int gehiketa (int a, int b)  
{  
    return a+b;  
}  
int kenketa (int a, int b)  
{  
    return a-b;  
}
```

Liburutegiak antolatzea

- Gatazka:

“fitxategia1.h”-an `#include<liburutegia1.h>` dago.

“fitxategia2.h”-an `#include<liburutegia1.h>` dago.

“fitxategia3.c”-n `#include<fitxategia1.h>` eta
`#include<fitxategia2.h>` lerroak daude.

- Ekiditzeko:

LIBURUTERIA_1_H fitxategia lehen aldiz sartzerakoan, definitu gabe egongo da. Beraz `#ifndef - #endif` multzoan sartuko da eta funtzioen eredu eta motak zehaztuko dira, baita LIBURUTEGIA_1_H bera ere. Bigarrenenez sartzerakoan, LIBURUTEGIA_1_H jadanik zehaztuta egongo da, eta ez da berriz sartuko `#ifndef - #endif` multzoan.

Liburutegi Estatikoa sortzeko

1. Liburutegia1-eko objektu modulua sortu:

gcc -o modulu1.o -c modulu1.c

- ar-k liburutegia sortzen du, baina edozein motako fitxategiak paketatzeko balio du (EZ objektu moduluak bakarrik). Gainera aukera desberdinak ditu: barnean zein fitxategi dauden ikusteko, fitxategi horiek ezabatu, ordezkatu, etab.

ar rv liborokor.a modulu1.o “r” aukerak ordezkatu edo barneratu egiten ditu objektu moduluak liburutegian. “v” aukerak mezuen bidez adierazten du “verbose”. Adibidez:

ar rv liborokor.a modulu1.o

ar: creating liborokor.a

a - modulu1.o

Liburutegi Estatikoa

- **ranlib**-ek artxibo hasieran bertan dauden funtzioen aurkibide sortzen du. Bertan funtzioen deskribapen eta estekak daude, liburutegi osoa irakurri gabe ebatzi ahal izateko identifikagailuak daude.

ranlib liborokor.a

Liburutegi Estatikoa

- **ranlib**-ek artxibo hasieran bertan dauden funtzioen aurkibide sortzen du. Bertan funtzioen deskribapen eta estekak daude, liburutegi osoa irakurri gabe ebatzi ahal izateko identifikagailuak daude.

ranlib liborokor.a

Liburutegi Estatikoa

- Aurkibidea ikusteko:

nm -s liborokor.a

Archive index:

gehiketa in modulu1.o

kenketa in modulu1.o

modulu1.o:

000000000000000000 T gehiketa

000000000000000014 T kenketa

Liburutegi Estatikoa

- Liburutegiko funtzio bat erabiltzen duen exekutagarria sortzeko:

Hau da aplikazioa.c:

```
#include <stdio.h> → /usr/include
```

```
#include "modulu1.h"
```

```
int main (void) {
```

```
    printf("%d",gehiketa(8,9));
```

```
    return 0;}
```

```
gcc -L . -l. -o aplikazioa aplikazioa.c -liborokor.a -static  
edo
```

```
gcc -L . -l. -o aplikazioa aplikazioa.c -lorokor -static  
./aplikazioa
```

Liburutegi Estatikoa

`gcc -L . -l. -o aplikazioa aplikazioa.c -lorokor -static`

`-I <path>` .h-ak non dauden jakiteko. (“/usr/include” ez den beste katalogo batean daudenean erabili.)

`-L<path>` .a edo .so non dauden jakiteko. (“/lib edo /usr/lib” ez den beste katalogo batean daudenean erabili)

`-lorokor` eginez, liburutegiaren izena adierazten dugu, lib aurrizkia eta .a luzapenik gabe (konpilatzean automatikoki jartzen baitira).

`-static` On systems that support dynamic linking, this prevents linking with the shared libraries.

Liburutegi Dinamikoa sortzeko

1. modulu1.c-eko objektu modulua sortu:

gcc -c modulu1.c -o modulu1.o -fPIC

2. Estekatzaileak liburutegia sortzen du:

ld -o liborokor.so modulu1.o -shared

-o liborokor.so aukerak liburutegiari eman nahi diogun izena adierazteko da.

-shared aukerak liburutegi bat (eta EZ exekutagarri bat) egin behar duela adierazteko da

-fPIC options enable "position independent code" generation, a requirement for shared libraries

Liburutegi Dinamikoa

3. Behin liburutegia sortuta, gure programa liburutegiarekin lotzeko:

```
gcc -o aplikazioa aplikazioa.c -l<path> -l<path>...  
-L<path> -L<path2>...-Bdynamic liburutegia1  
liburutegia2
```

Adb: **gcc -o aplikazioa aplikazioa.c -l. -L . -
Bdynamic liborokor.so**

-Bdynamic aukerarekin liborokor.a hartu beharrean liborokor.so hartzen du.

Liburutegi Dinamikoa

4. Behin exekutagarria konpilatuta dugunean, azken pauso bat falta da. Programak liburutegi dinamikoan dagoan funtzio bati deitzerakoan, gure liburutegi dinamikoak bilatzera nora joan behar den sistemari esango diogu, horretarako, LD_LIBRARY_PATH aldagaia moldatuko dugu.

`LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/kepa/bin`

Eta ondoren idatzi : `export LD_LIBRARY_PATH`

5. Orain exekutatu dezakegu: `./aplikazioa`

Dena mainean

```
$gedit geometria.c
#include <stdio.h>
int karratu(int x);
int kubo (int x);
int main(){
int aldea;
printf("Alde bat sartu \n");
scanf("%d",&aldea);
printf("Kuboa %d da
\n",kubo(aldea));
```

```
printf("Karratua %d da
\n",karratua(aldea));
return 0;
}
int karratu(int x)
{return x*x;}
int kubo (int x)
{return x*x*x;}
```

gcc geometria.c -o geometria

Adibidea:

```
$gedit define.h  
#ifndef _DEFINE_H  
#define _DEFINE_H  
#define PI 3.1415  
#endif
```

```
$gedit bolumena.h  
#ifndef _BOLUMENA_H  
#define _BOLUMENA_H  
int kubo (int x);  
float esfera (int x);  
#endif
```

```
$gedit azalera.h  
#ifndef _AZALERA_H  
#define _AZALERA_H  
int karratu (int x);  
float zirkulu (int x);  
#endif
```

Adibidea

```
$gedit azalera.c  
#include "define.h"  
int karratu(int x)  
{return x*x;}  
float zirkulu(int x)  
{return PI*x*x;}
```

```
$gedit bolumena.c  
#include "define.h"  
int kubo (int x)  
{return x*x*x;}  
float esfera (int x)  
{return 4*PI*x*x*x/3;}
```


Adibidea

```
$gedit geometria.c
#include <stdio.h>
#include "azalera.h"
#include "bolumena.h"
int main(){
int aldea,erradio;
printf("Alde bat sartu \n");
scanf("%d",&aldea);
```

```
printf("erradio bat sartu\n");
scanf("%d",&erradio);
printf("Kuboa %d da
\n",kubo(aldea));
printf(" Esfera %f da
\n",esfera(erradio));
printf("Karratua %d da
\n",karratua(aldea));
printf("zirkulu %f da
\n",zirkulu(erradio));
return 0;
}
```

Adibidea: Liburutegi gabe

```
gcc -l. -c bolumena.c -o bolumena.o
```

```
gcc -l. -c azalera.c -o azalera.o
```

```
gcc -l. -c geometria.c -o geometria.o
```

```
gcc -l. bolumena.o azalera.o geometria.o -o  
geometria
```

Adibidea: liburutegi estatikoa

```
gcc -I. -c bolumena.c -o bolumena.o
```

```
gcc -I. -c azalera.c -o azalera.o
```

```
ar rv liborokor.a bolumena.o azalera.o
```

```
ranlib liborokor.a
```

```
gcc -I. -L. geometria.c liborokor.a -o geometria  
-static
```

Adibidea: liburutegi dinamikoa

```
gcc -fPIC -c bolumena.c -o bolumena.o
gcc -fPIC -c azalera.c -o azalera.o
ld -shared bolumena.o azalera.o -o liborokor.so
gcc -Bdynamic -l. -o geometria geometria.c liborokor.so
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/kepa/lana
export LD_LIBRARY_PATH
./geometria
```