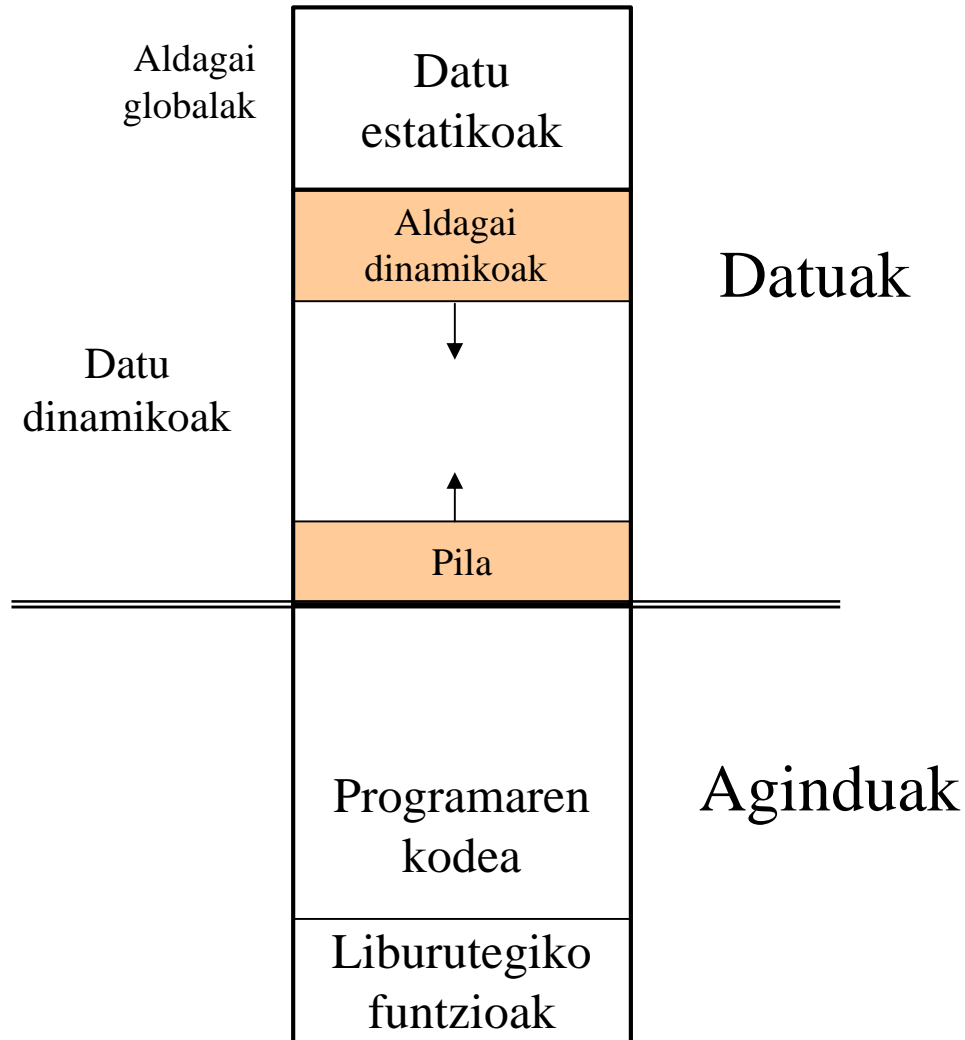


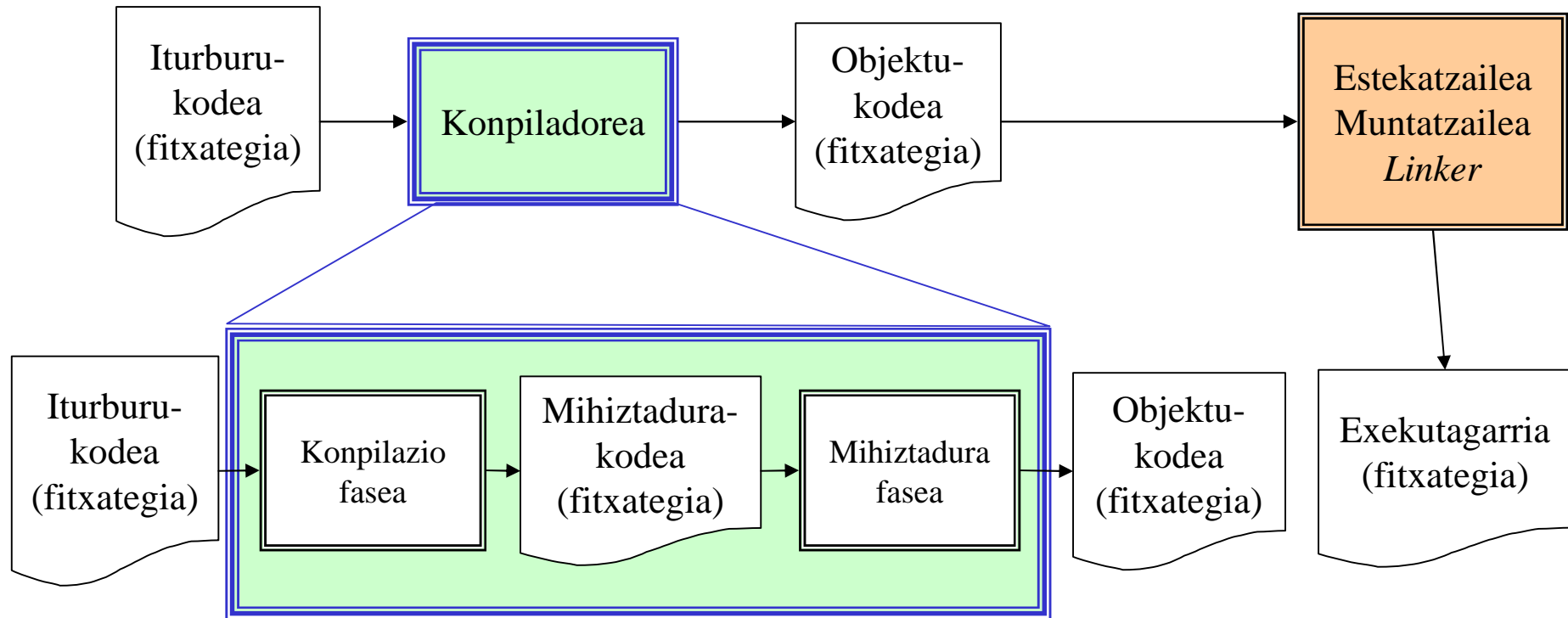
5. Gaia. Programen karga eta kokapena memorian

1. Iturburu kodea. Konpilazioa.
2. Objektu liburutegiak. Muntaia/Estekatzea.
3. Programak memorian kokatzeko moduak.
4. Estekatze dinamikoa.

Programa baten egitura memorian

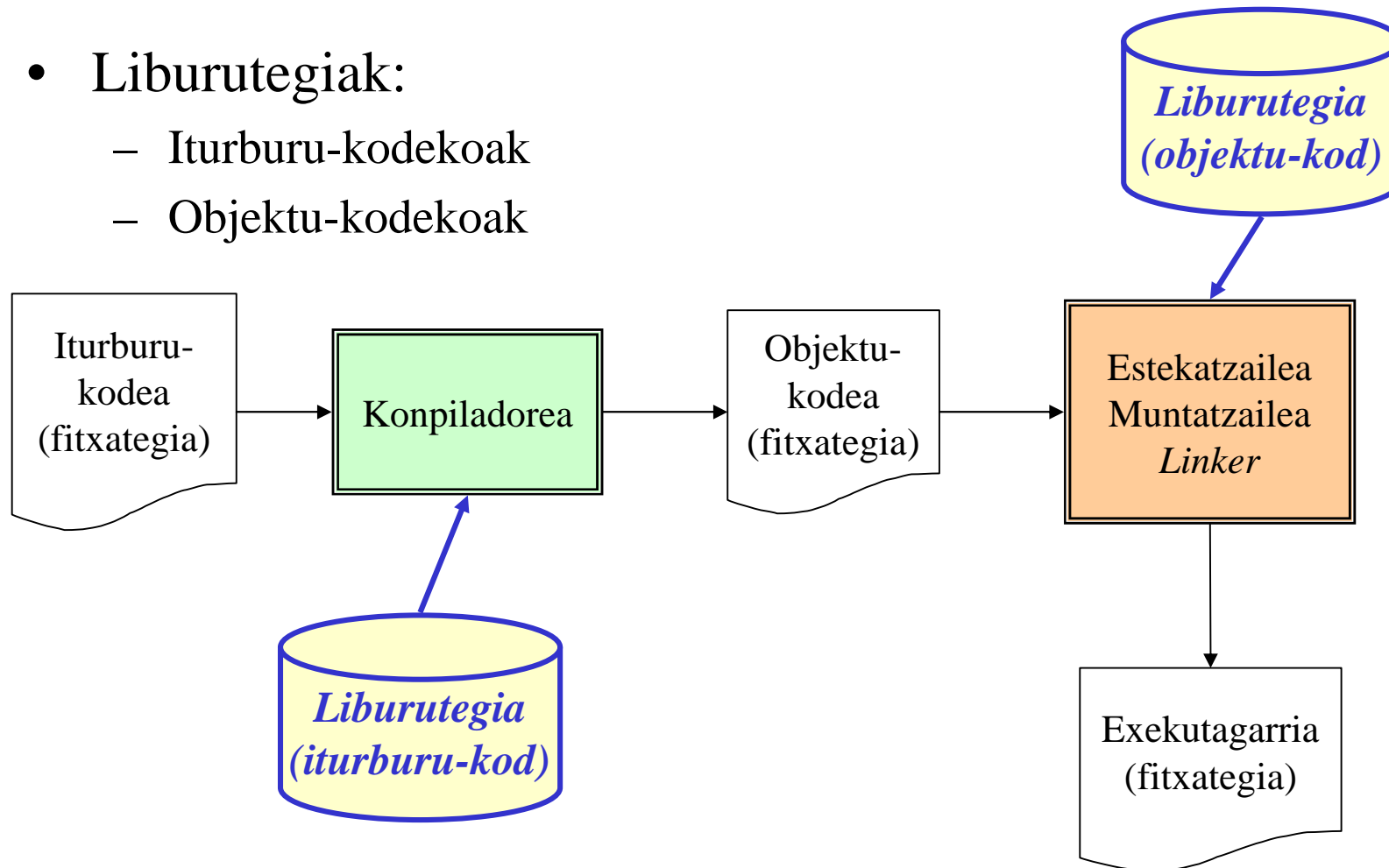


Konpilazioa



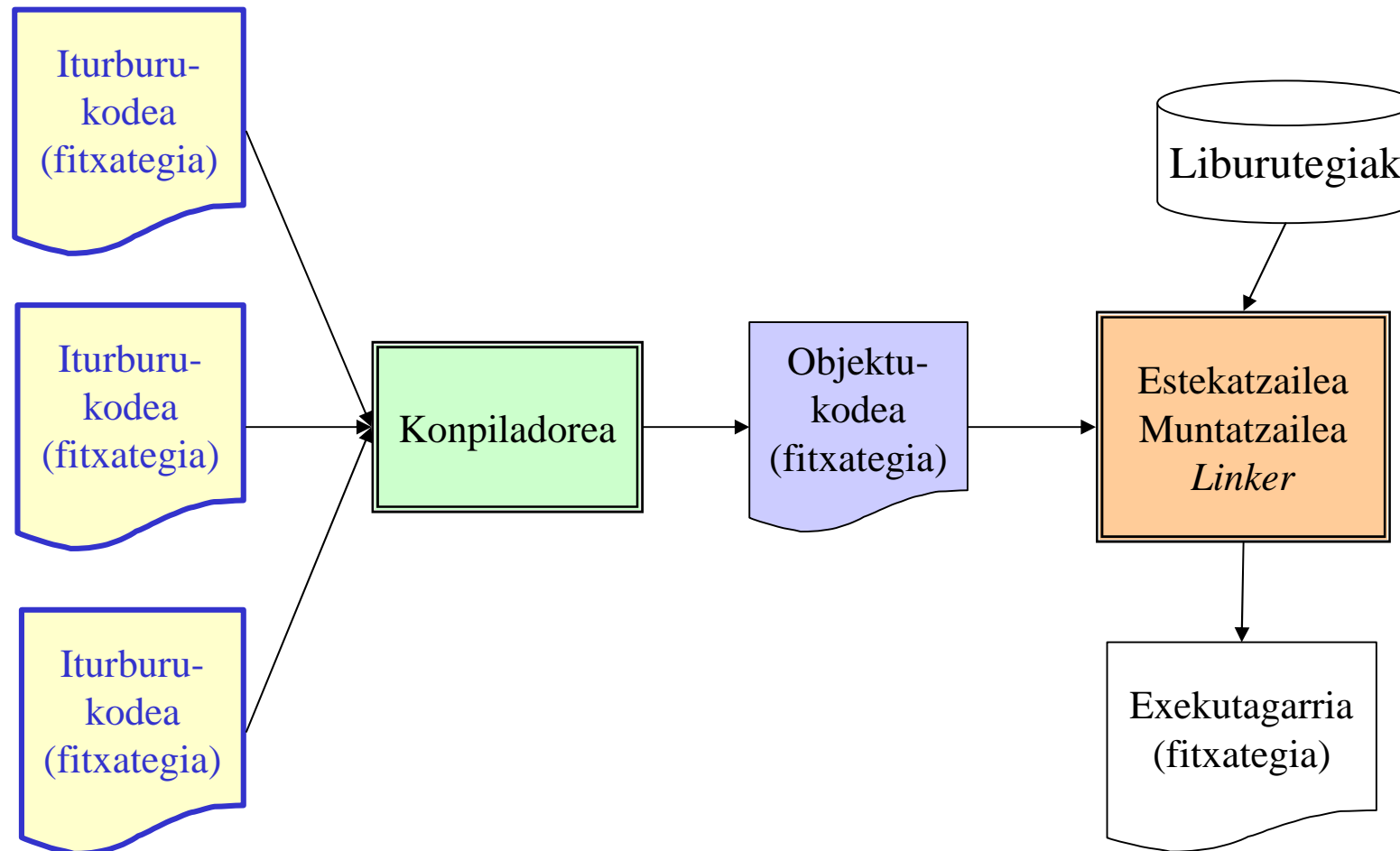
Konpilazioa

- Liburutegiak:
 - Iturburu-kodekoak
 - Objektu-kodekoak



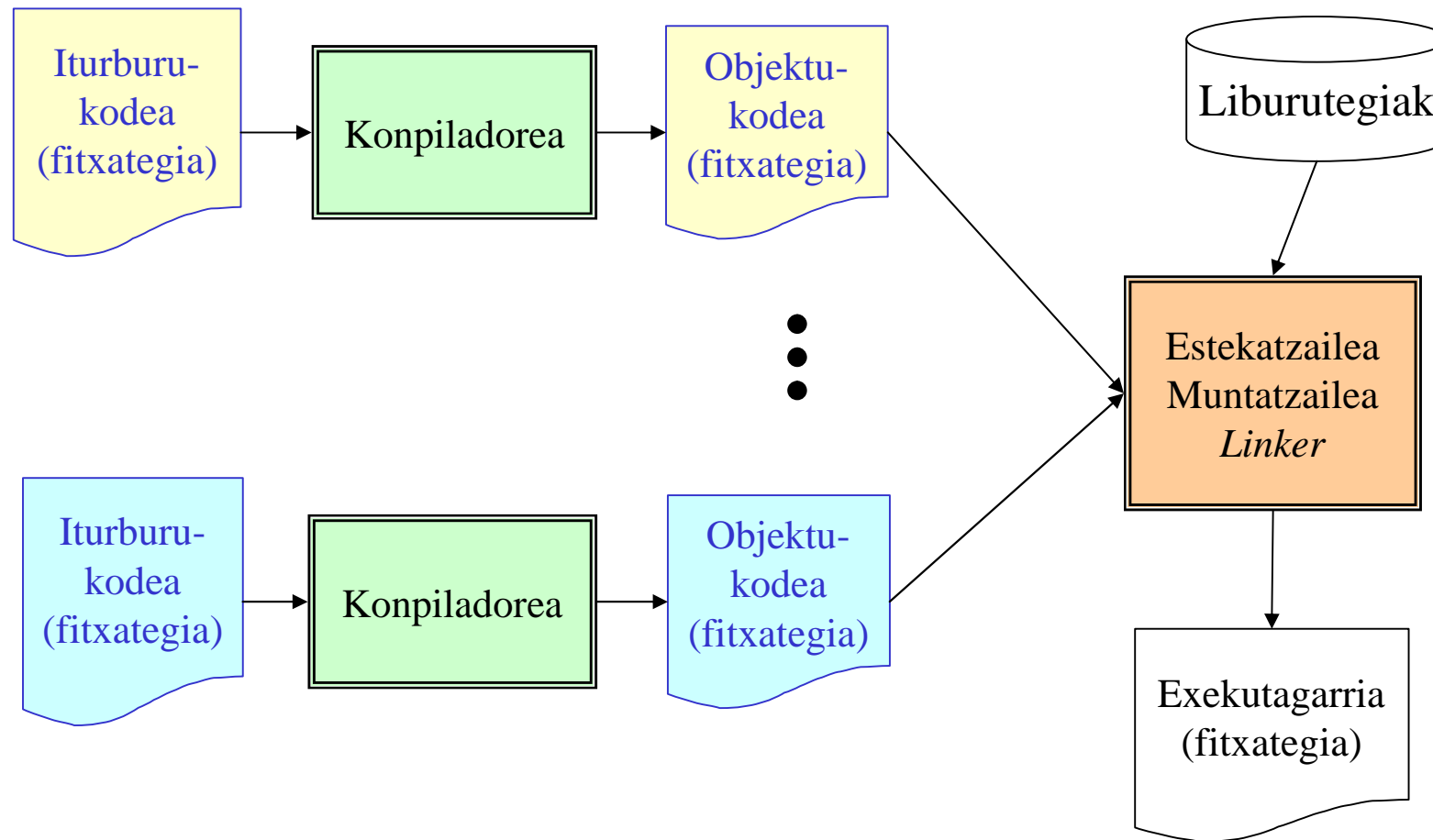
Konpilazioa

- Hainbat iturburu \Rightarrow Objektu modulu bakarra



Konpilazioa

- Hainbat iturburu \Rightarrow Hainbat objektu modulu



```
43-sisd00% gcc -v kopiatu.c -o kopiatu
```

```
Reading specs from /usr/local/lib/gcc-lib/sparc-sun-solaris2.6/2.8.1/specs  
gcc version 2.8.1
```

```
/usr/local/lib/gcc-lib/sparc-sun-solaris2.6/2.8.1/cpp -lang-c -v -undef -  
D__GNUC__=2 -D__GNUC_MINOR__=8 -Dsparc -Dsun -Dunix -D__svr4__ -D__SVR4 -  
D__sparc__ -D__sun__ -D__unix__ -D__svr4__ -D__SVR4 -D__sparc -D__sun -D__unix -  
Asystem(unix) -Asystem(svr4) -D__GCC_NEW_VARARGS__ -Acpu(sparc) -Amachine(sparc)
```

```
kopiatu.c /var/tmp/ccmTaq5y.i
```

```
GNU CPP version 2.8.1 (sparc)
```

```
#include "... " search starts here:
```

```
#include <...> search starts here:
```

```
/usr/local/include
```

```
/usr/local/sparc-sun-solaris2.6/include
```

```
/usr/local/lib/gcc-lib/sparc-sun-solaris2.6/2.8.1/include
```

```
/usr/include
```

```
End of search list.
```

```
/usr/local/lib/gcc-lib/sparc-sun-solaris2.6/2.8.1/cc1 /var/tmp/ccmTaq5y.i -  
quiet -dumpbase kopiatu.c -version -o /var/tmp/ccmTaq5y.s
```

```
GNU C version 2.8.1 (sparc-sun-solaris2.6) compiled by GNU C version 2.8.1.
```

```
/usr/ccs/bin/as -V -Qy -s -o /var/tmp/ccmTaq5y1.o /var/tmp/ccmTaq5y.s
```

```
/usr/ccs/bin/as: WorkShop Compilers 5.0 Alpha 03/27/98 Build
```

```
/usr/ccs/bin/ld -V -Y P,/usr/ccs/lib:/usr/lib -Qy -o kopiatu
```

```
/usr/local/lib/gcc-lib/sparc-sun-solaris2.6/2.8.1/crt1.o /usr/local/lib/gcc-  
lib/sparc-sun-solaris2.6/2.8.1/crti.o /usr/ccs/lib/values-Xa.o
```

```
/usr/local/lib/gcc-lib/sparc-sun-solaris2.6/2.8.1/crtbegin.o -
```

```
L/usr/local/lib/gcc-lib/sparc-sun-solaris2.6/2.8.1 -L/usr/ccs/bin -L/usr/ccs/lib
```

```
-L/usr/local/lib /var/tmp/ccmTaq5y1.o -lgcc -lc -lgcc /usr/local/lib/gcc-
```

```
lib/sparc-sun-solaris2.6/2.8.1/crtend.o /usr/local/lib/gcc-lib/sparc-sun-
```

```
solaris2.6/2.8.1/crtn.o
```

```
ld: Software Generation Utilities - Solaris/ELF (3.0)
```

7

```
44-sisd00%
```

Goi-mailako lengoaiak (GML)

- Konpiladorea
 - Goi-mailako lengoaiako *agindu 1* => makina-lengoaiako *n agindu*
- Itzulpena + Optimizazioa
 - + eraginkorra
 - + trinkoa
- Arazoa eragigaien helbideak itzultzean: aginduak eta aldagaiak zein memoria-helbidetan kokatu?
 - Konpiladoreak hasierako karga-helbidea ezagutzen du (*Konpilazio-denborako birkokapena*).
 - Konpiladoreak hasierako karga-helbidea 0 helbidea dela suposatzen du. Ondoren, programa memorian kargatzerakoan (kargatzaileak) karga-helbide errealarekin birkokatzen ditu helbide guztiak (*Karga-denborako birkokapena*).
- *Birkokapen estatikoa*
 - Birkokapen-taula

Iturburu-kodea. Konpilazioa

Mihiztadura-kodea

Iturburu-programa
goi-mailako lengoaian (C)

```
...
int Globala1, x, Emainza;

main()
{
    Globala1 = 35;
    x = 3;
    Emainza = Globala1 + x*x;
    idatzi(Emainza);
}

void idatzi(int E)
{
    printf("Emaitza: %d izan da\n", E);
}
```

*Helbide
logikoak*

Konpiladorea

```
0: Globala1:
4: x:
8: Emainza:
   PROC main
12: PUSH r30           ;; main
16: MOV r30, r31
20: MOV r1, 35
24: ST r1, @Globala1
28: MOV r1, 3
32: ST r1, @x
36: LD r2, @Globala1
40: LD r3, @x
44: MULT r4, r3, r3
48: ADD r5, r2, r4
52: ST r5, @Emaitza
56: PUSH r5
60: CALL idatzi
...
128: RET

   PROC idatzi
132: PUSH r30           ;; idatzi
...
252: RET
```

Mihiztadura-kodea

Bigarren fasea

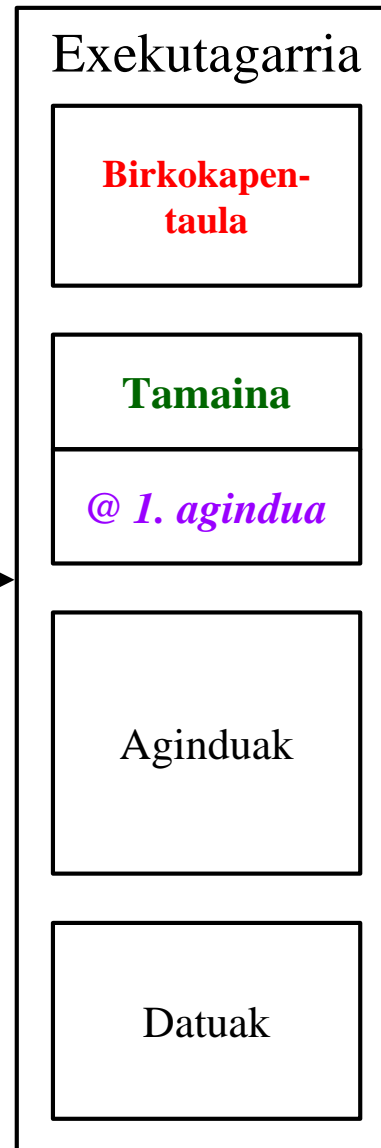
```
0: Globala1:
4: x:
8: Emaiza:
   PROC main
12: PUSH r30 ;; main
16: MOV r30, r31
20: MOV r1, 35
24: ST r1, @Globala1
28: MOV r1, 3
32: ST r1, @x
36: LD r2, @Globala1
40: LD r3, @x
44: MULT r4, r3, r3
48: ADD r5, r2, r4
52: ST r5, @Emaiza
56: PUSH r5
60: CALL idatzi
...
128: RET

   PROC idatzi
132: PUSH r30 ;; idatzi
...
252: RET
```

Mihiztatzailea

Exekutagarrien formatua:

a.out,
COFF,
ELF,
...

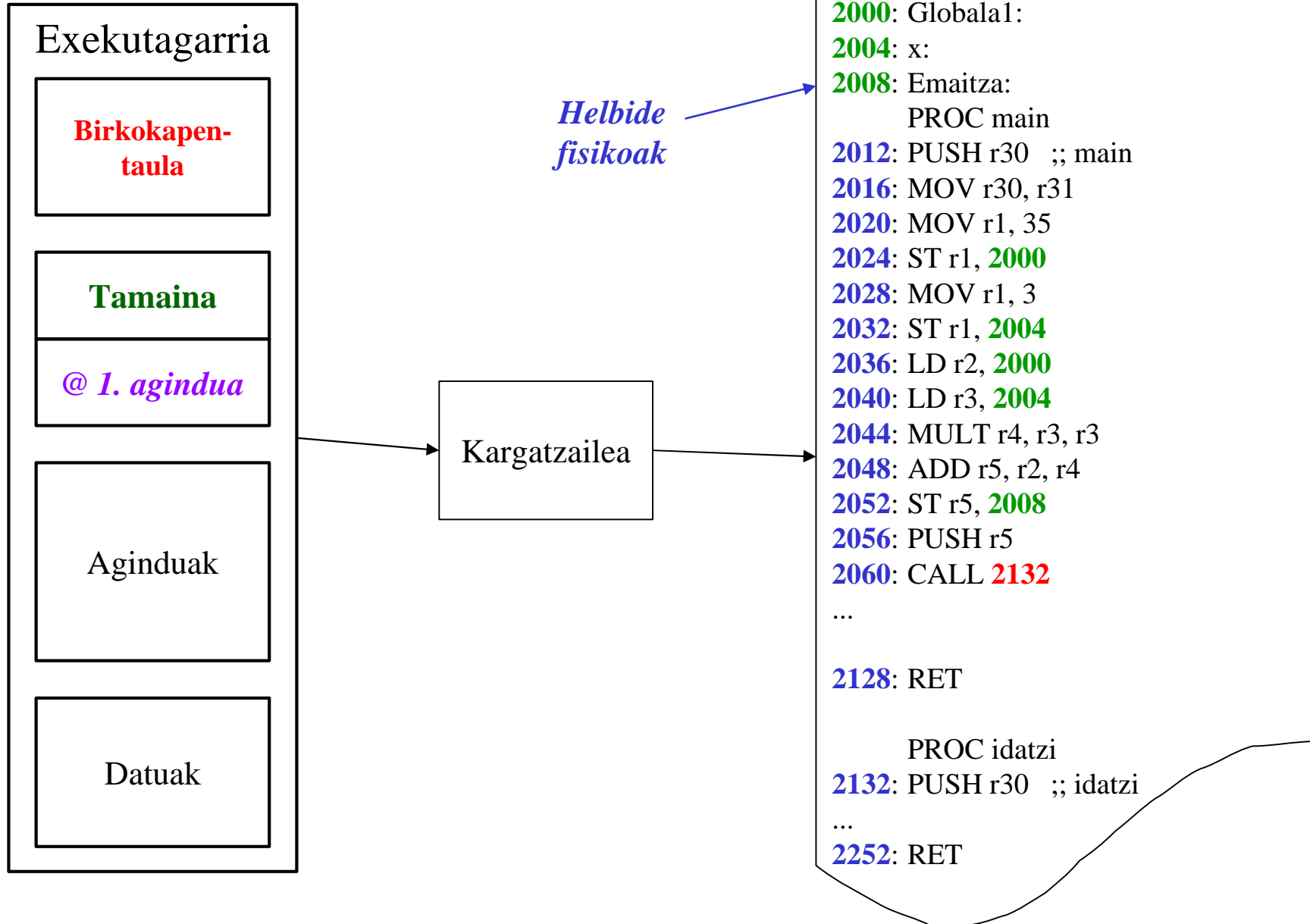


Kargatzailea

- Kargatzailearen betebeharrak:
 1. Fitxategi exekutagarria irakurri
 2. Memorian kokatu:
 - Memorian behar beste tamainako zati libre eta jarraian bilatu
 - Programaren kodea kopiatu
 - Hasieratutako datuak kopiatu
 - Datu dinamikoentzako tokia erreserbatu (pila, memoria dinamikoa)
 3. Programako lehen agindua abiatu:
 - CPU-aren erregistroak hasieratu (PC, PSW, SP...)

Karga-fasea

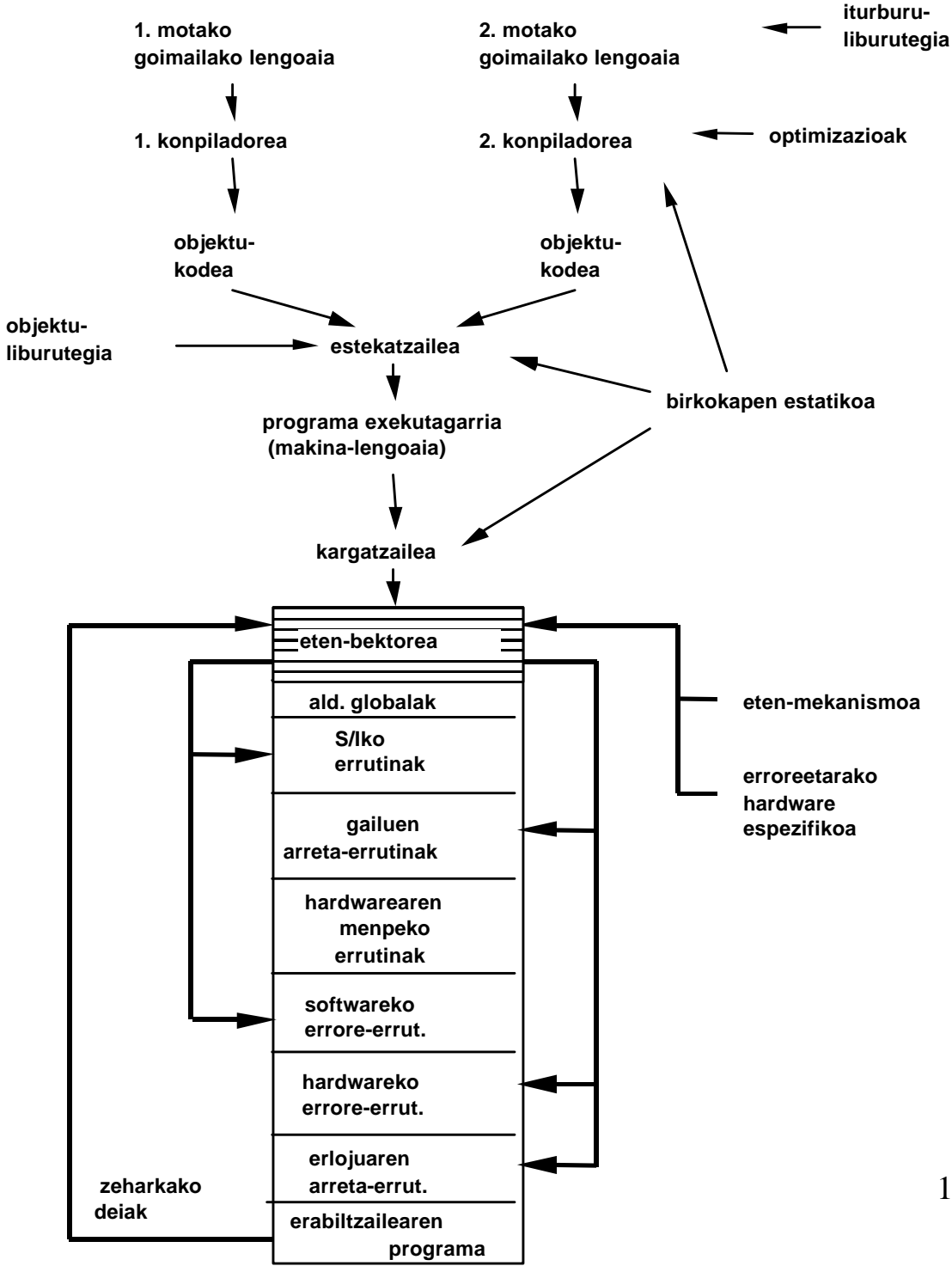
Mihiztadura-kodea



Liburutegiak: errutinen berrerabilpena

- Helburuak: birprogramazio lana minimizatu, erroreak minimizatu
- Goi-mailako lengoaiako funtzioen liburutegiak (iturburu-liburutegiak)
 - Ez dira birprogramatu behar. Baina birkonpilatu behar dira
- Objektu-kodeko funtzioen liburutegiak (dagoeneko konpilatuak)
 - Konpilazio banatua
 - Iturburu-programan agertzen ez diren errutinen aipamenak agertzen dira
 - Errutinek programa deitzaileak dauzkan aldagai globalak erreferentzia ditzakete
 - **EBATZI GABEKO ERREFERENTZIEN** arazoa
- Muntatzailea – Estekatzailea (*Linker*)
 - Ebatzi gabeko erreferentziak konpontzen ditu, exekutagarria eraikiz
 - Goi-mailako lengoai desberdinen konbinaketa
 - Egitura bereko objektu-kodea
 - Teknika berdinak errutinei parametroak pasatzeko eta aldagai globalak atzitzeko
- ***Estekatze-denborako birkokapena***

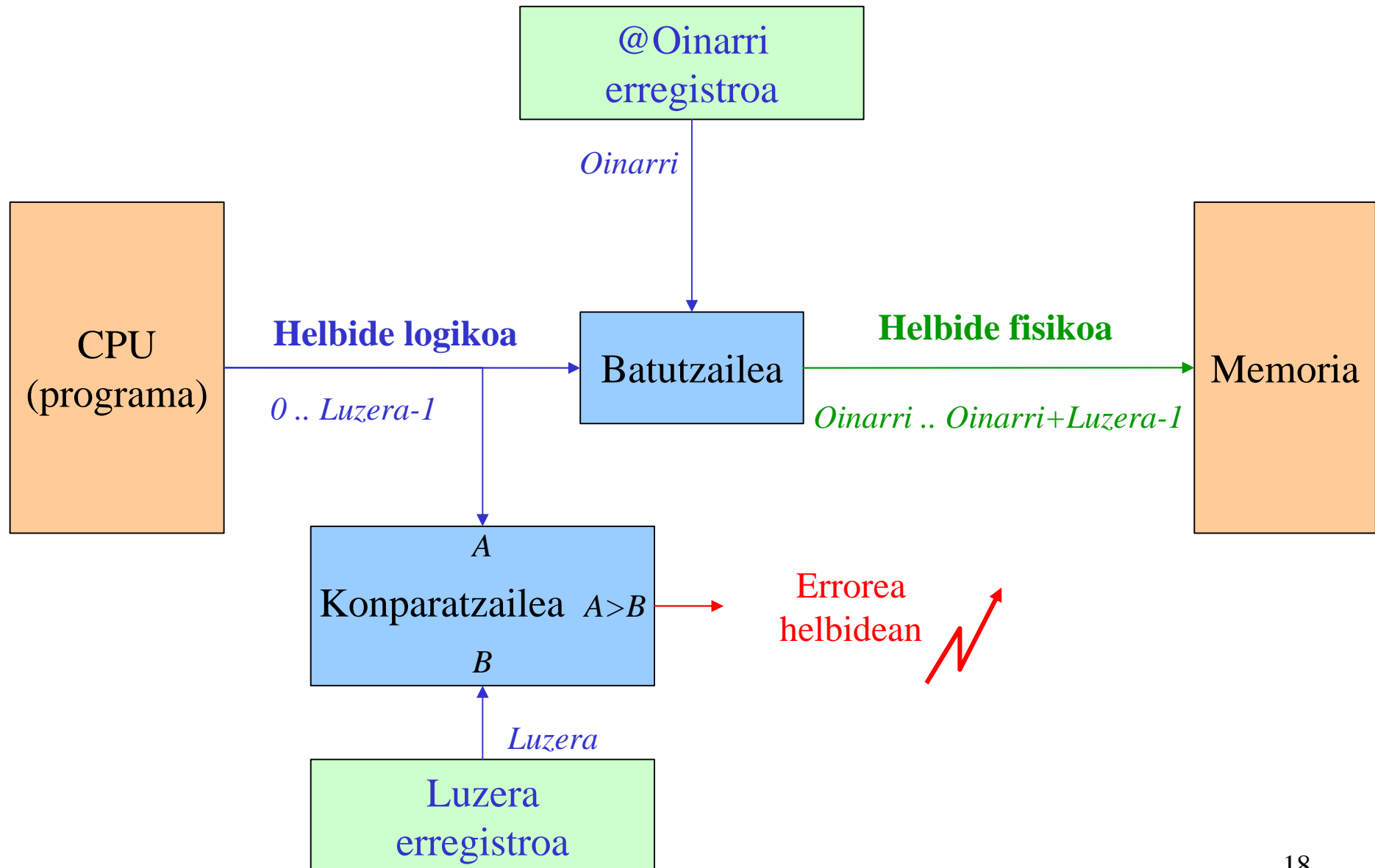
Laburpena



Programak memorian kokatzeko moduak

- Programa bat baino gehiago memorian
 - EZ JARRAIAN
 - EZ OSORIK
- Birkokapen estatikoa *versus* birkokapen dinamikoa

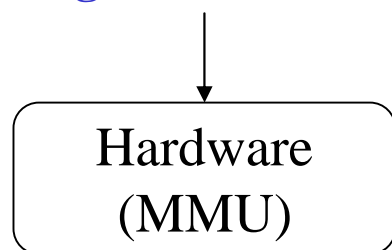
Memoriaren babeserako hardware-eskema



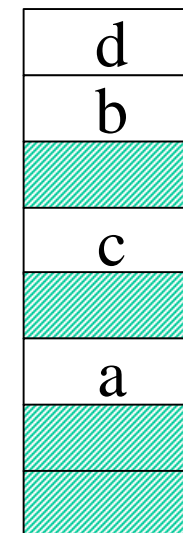
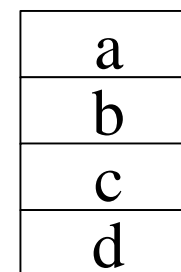
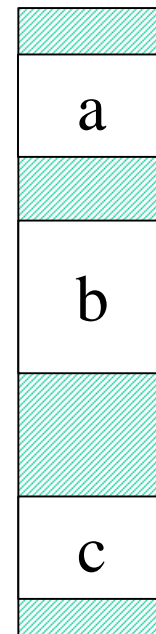
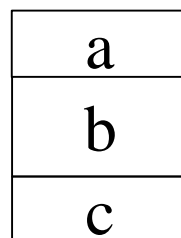
Programa EZ JARRAIAN eta EZ OSORIK

❑ EZ JARRAIAN

Alegiazko helbidea



Helbide fisikoa

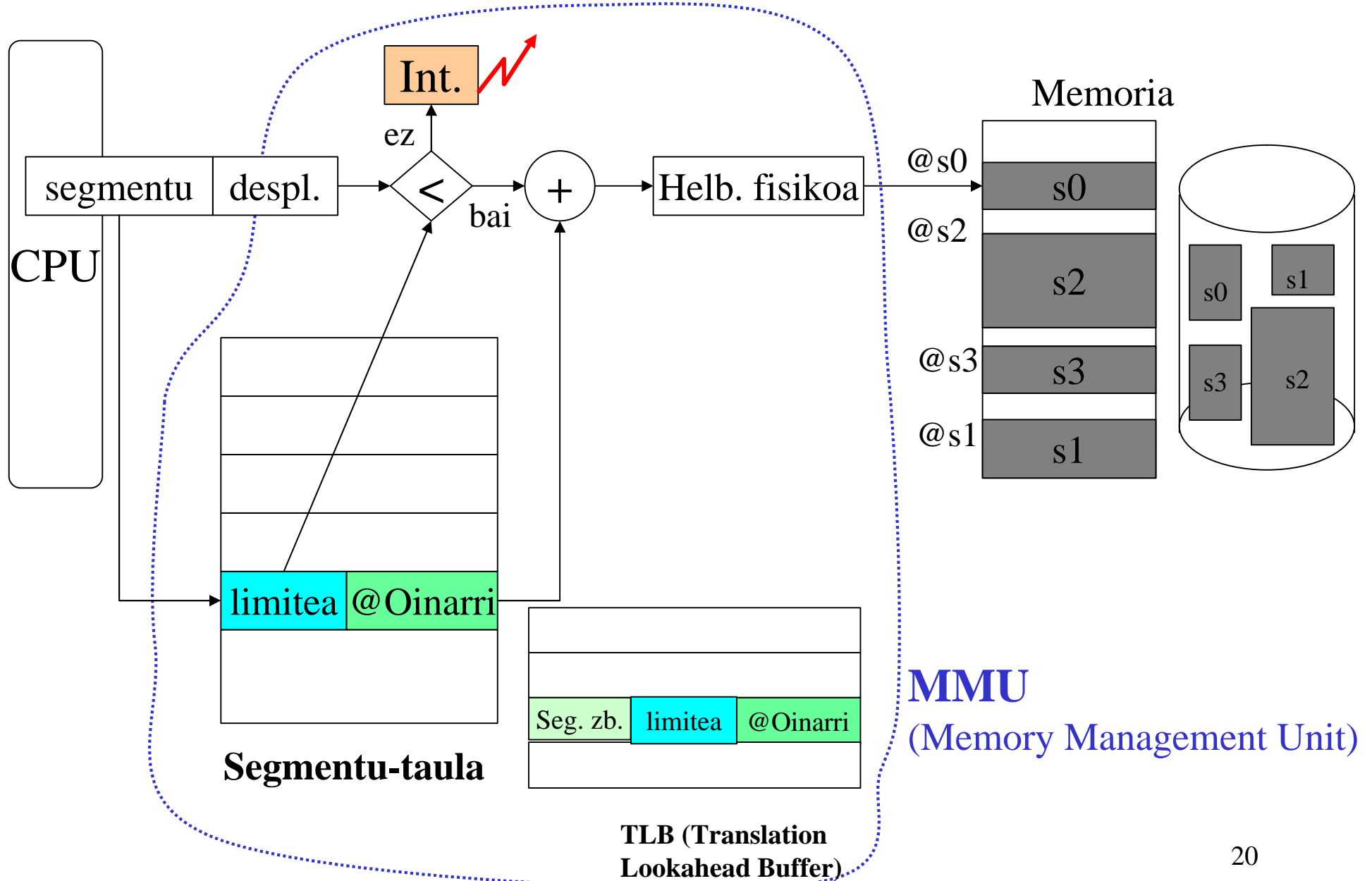


➤ Programa zatitan banatu. Bi teknika:

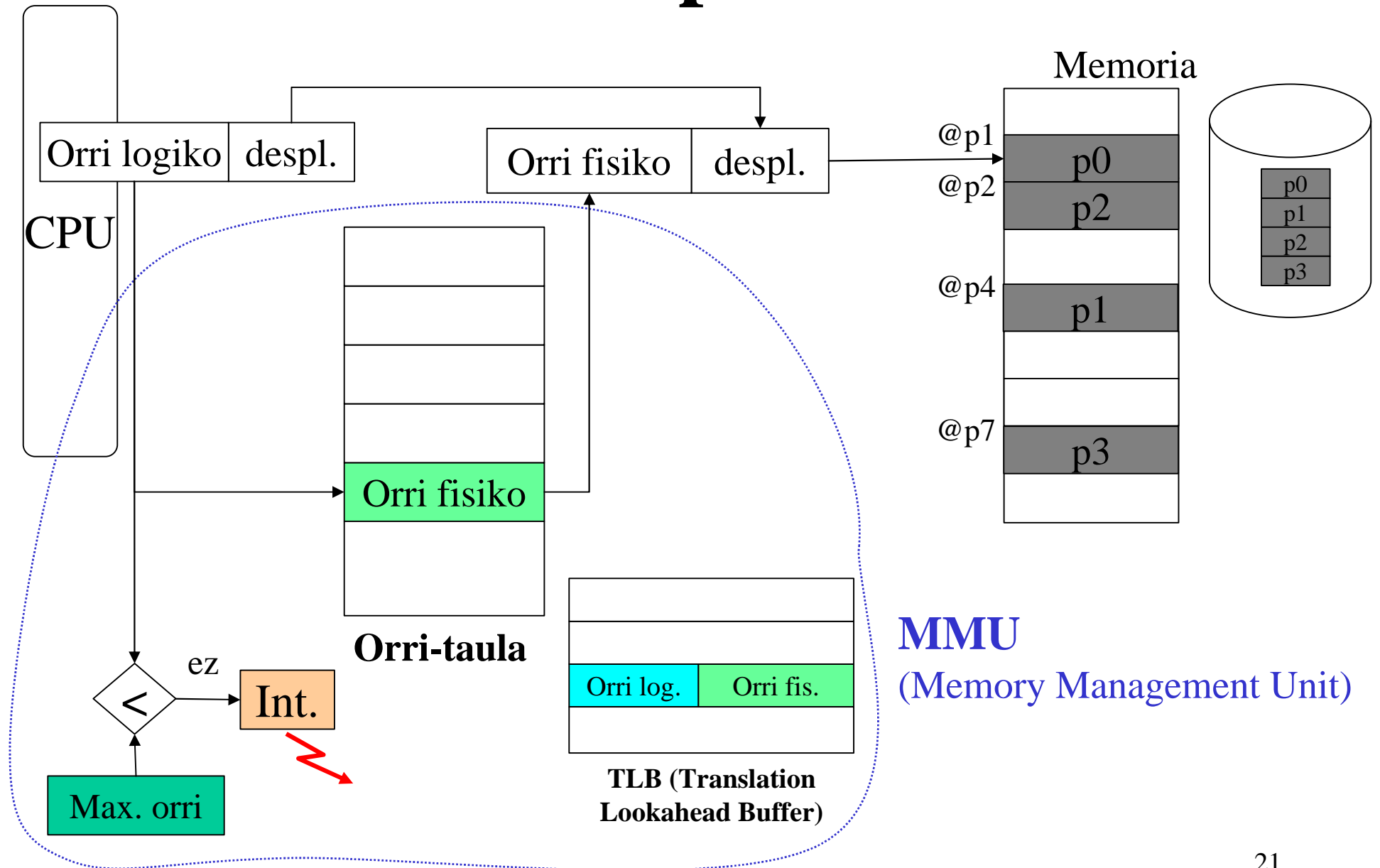
➤ Tamaina desberdineko zatiak: **SEGMENTAZIOA**
Konpiladoreak egiten du

➤ Tamaina berdineko zatiak: **ORRIKAPENA**
Sistema Eragileak egiten du

Segmentazioa



Orrikapena

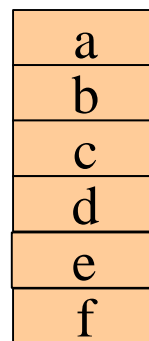


Programa EZ JARRAIAN eta EZ OSORIK

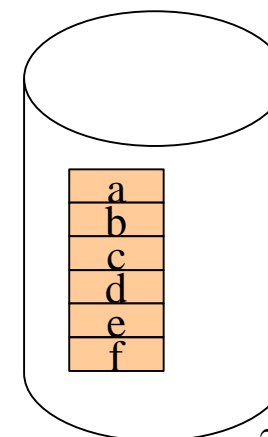
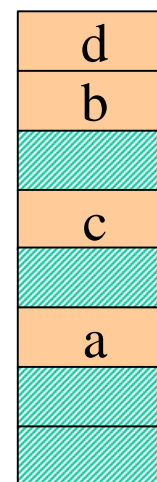
❑ EZ OSORIK

- Alegiazko memoria (diskoan, *swap area*)
- Orrikapena edo segmentazioarekin konbinatuz
- Atzipen-hutsegitea (**orri-hutsegitea**): itzulpena egiteko unean, alegiazko helbide bati dagokion helbide fisikoa ez existitzea
 - Hutsegiteen arreta-errutina: falta den orria diskotik memoriara pasa
 - Ordezkatze-politikak - Hiper-orrikapena (*trashing*): eraginkortasunaren jaitsiera
 - Karga aurreratua / eskaripean

Programa



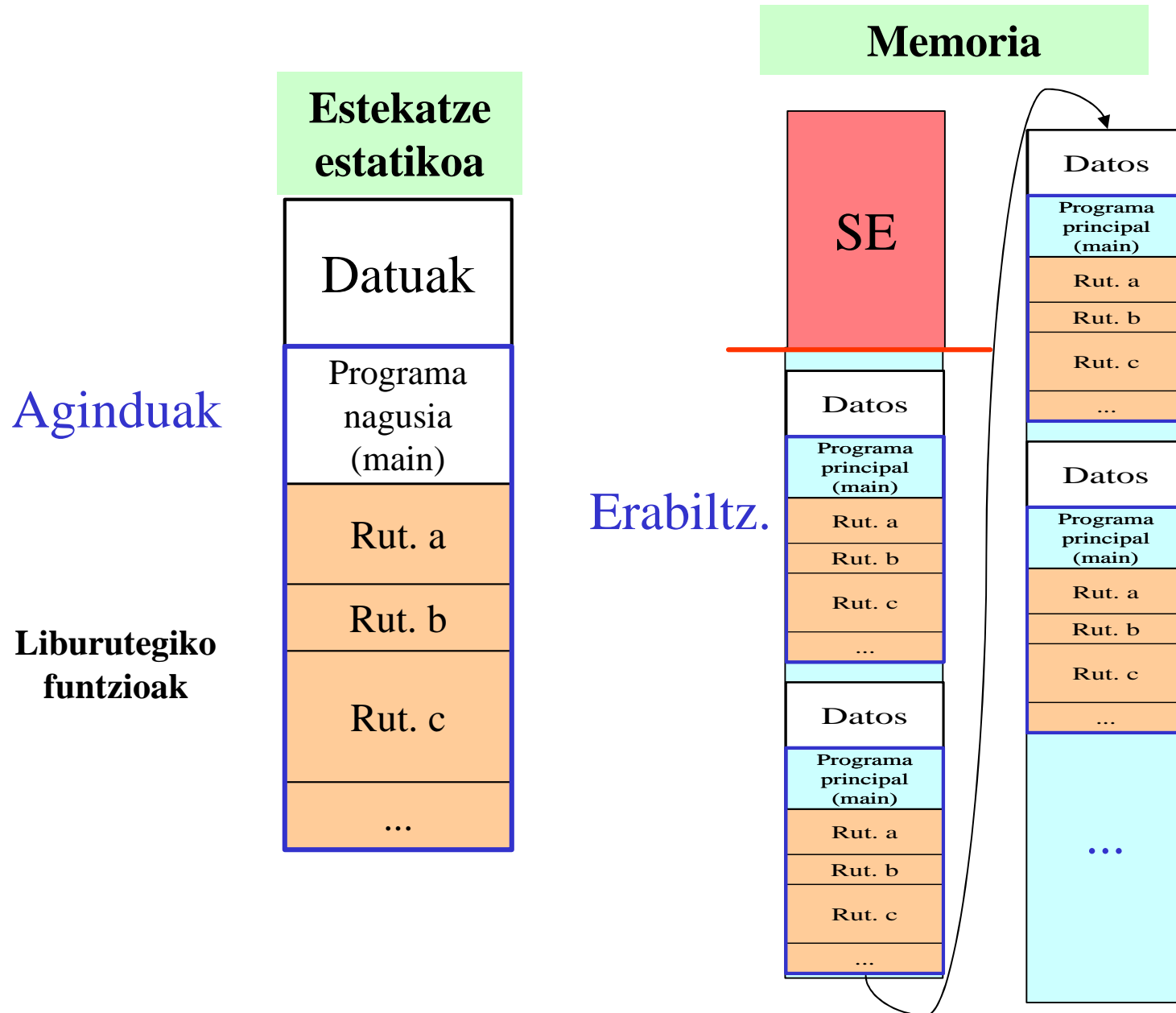
Memoria



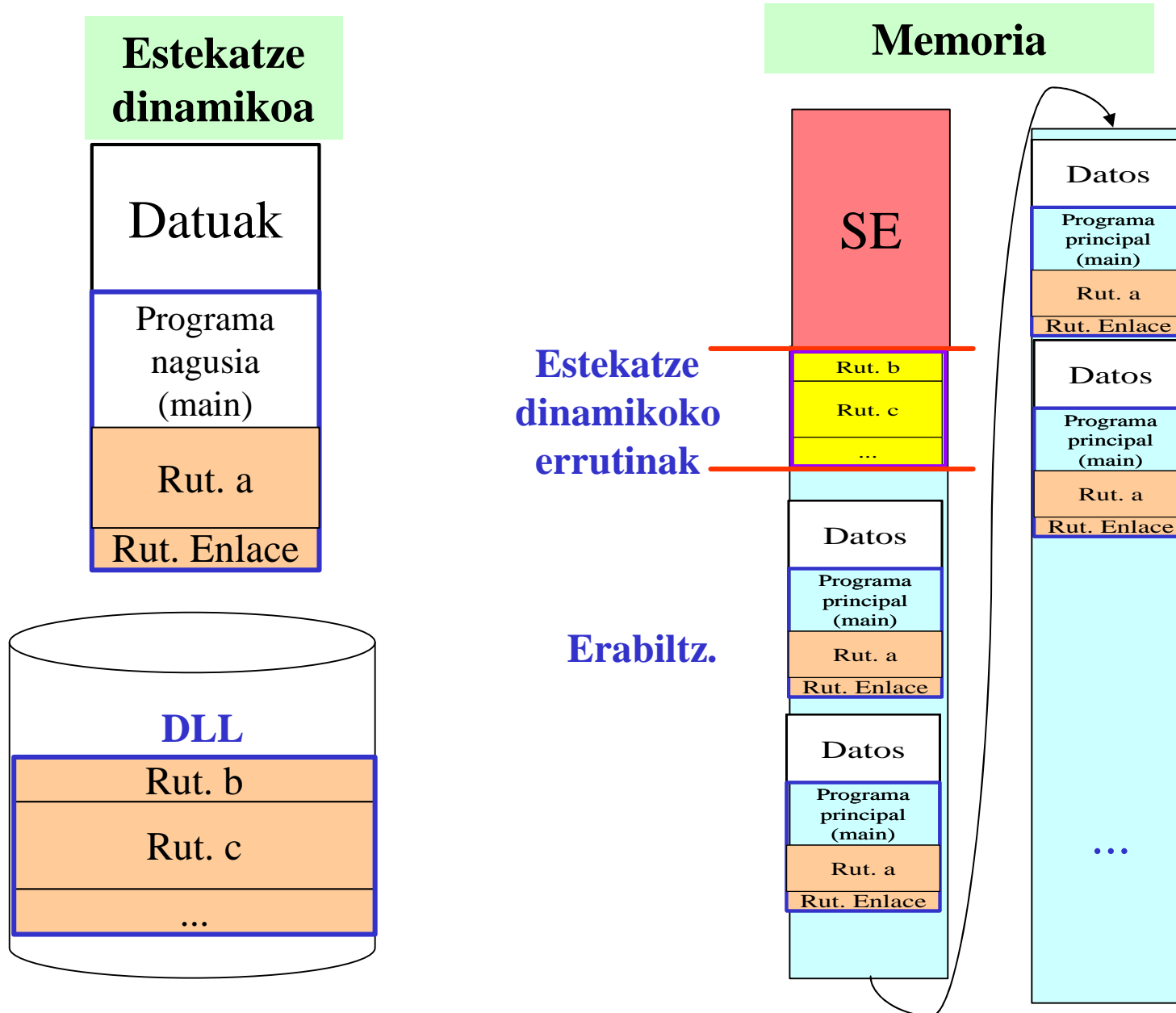
Estekatze dinamikoa

- Sistema Eragilearen errutinetarako pentsaturiko errutinen berrerabilgarritasuna edozein motako errutinetara zabaldu
- Espazio-aurrezpena, programek gutxiago okupatzen dute, bai memorian eta bai diskoan
- Estekatze dinamikoko liburutegia. Programetan mota honetako funtzioei deitzeko, errutina berezia erabiltzen da
 - Estekatzailak programa exekutagarrian errutinak zuzenean sartu beharrean sistema-dei berezi bat ipintzen du
 - Exekuzio-denboran kargatzen dira, soilik kargaturik gabe badaude
 - programa kargatzen denean / funtzioari deitzen zaionean
- Programak aldatu/eguneratu daitezke birkonpilatu gabe
- SEaren errutinetara aplikatu daiteke
 - SEaren zati desberdinak aldatu daitezke sistema gelditu gabe
- Memoria fisikoa baino handiagoak diren programak exekutatu daitezke (alegiazko memoria mekanismoa)

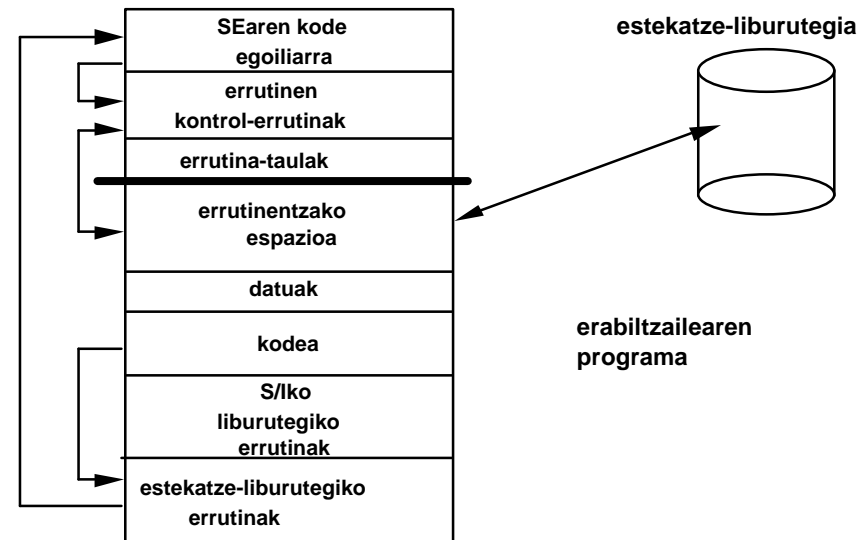
Estekatz e estatikoa



Estekatzeko dinamikoak



Estekatz dinamikoa



estekatz-liburategia

```
estekatz_lib_errut (char *izena)
{ par1=izena;
  sistema-deia(KONTROL_ERRUT);
  if (par2==ERRORERIK_EZ)
    (*par2)();
  else
    erroreak(ESTEKATZEA,par2);
}
```

aldagai globalak

```
int (*par2)(); char *par1;
```

kontrol-errutina

```
errutinen_kontrol_errut ()
{ if ((par2=badago(par1))==EZ)
  par2 = kargatu_ordezkatu_errut(par1);
}
```

estekatz-liburategia

```
izena: errut_dinam_1
```

```
/* errutinaren kodea */
```

```
izena: errut_dinam_2
```

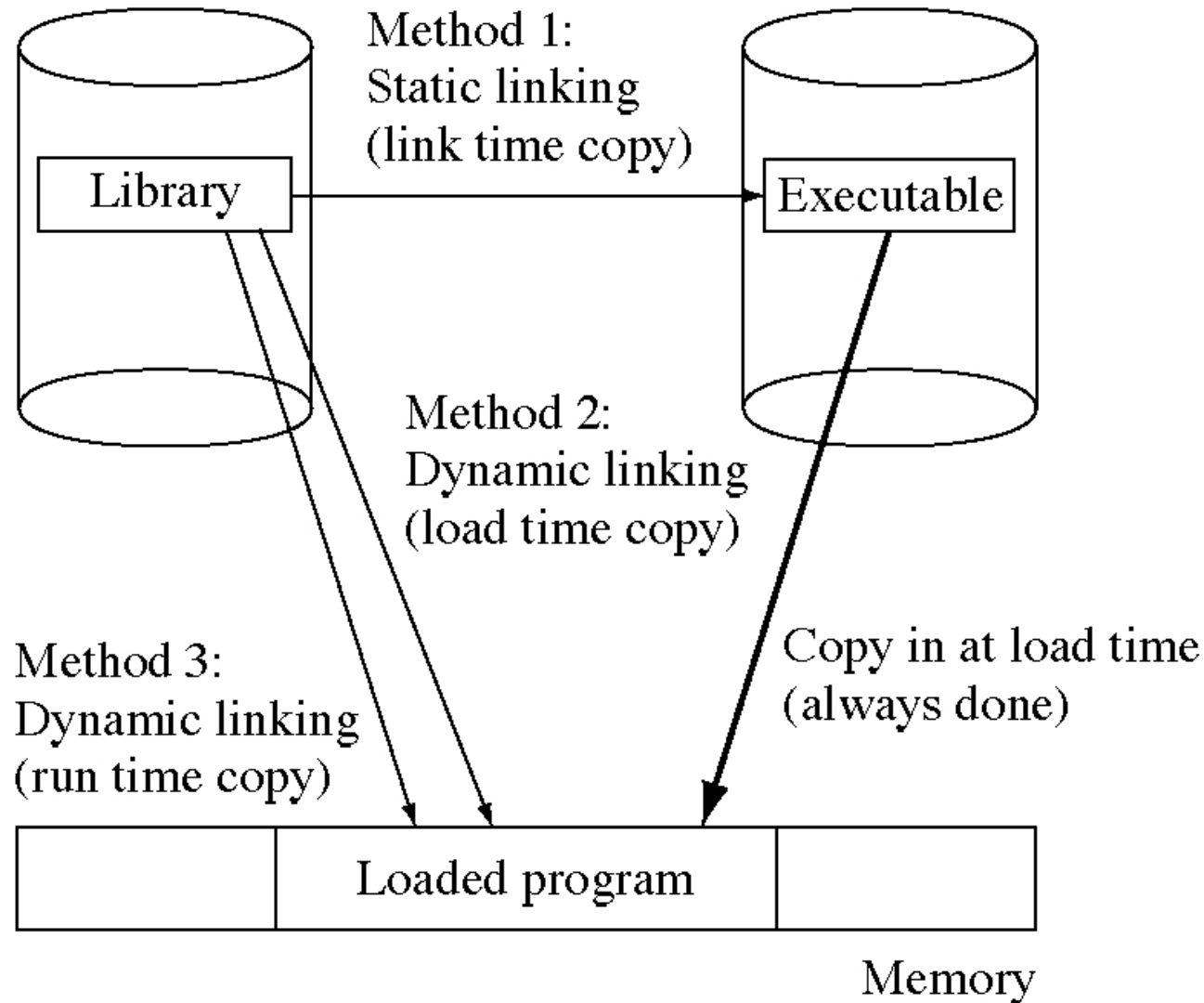
```
/* errutinaren kodea */
```

```
beste errutinak
```

erabiltzailearen programa

```
/* programaren kodea */
estekatz_lib_errut("errut_dinam_1");
/* programaren kodea */
estekatz_lib_errut("errut_dinam_2");
/* programaren kodea */
```

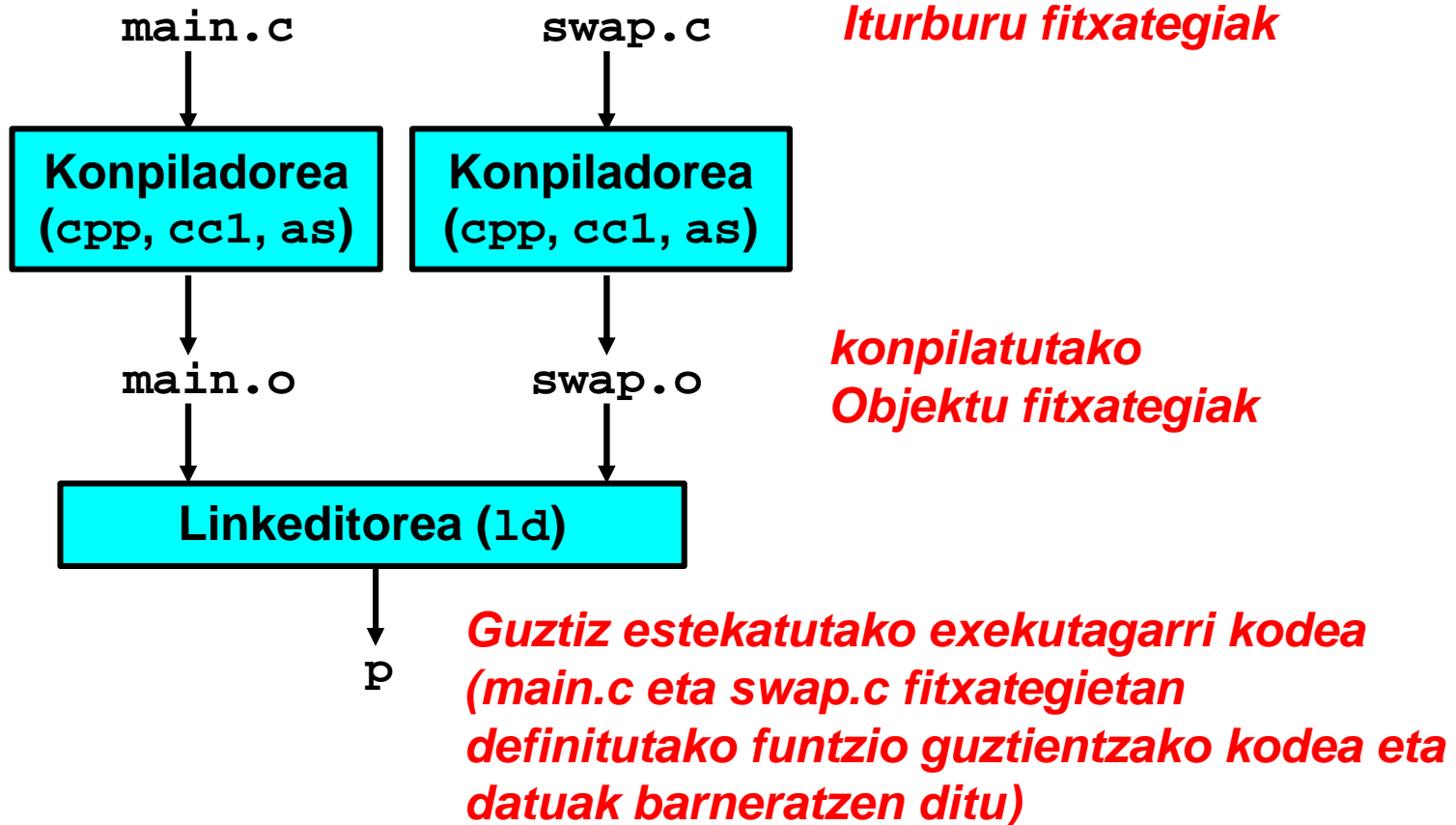
Estekatze estatikoa eta dinamikoa



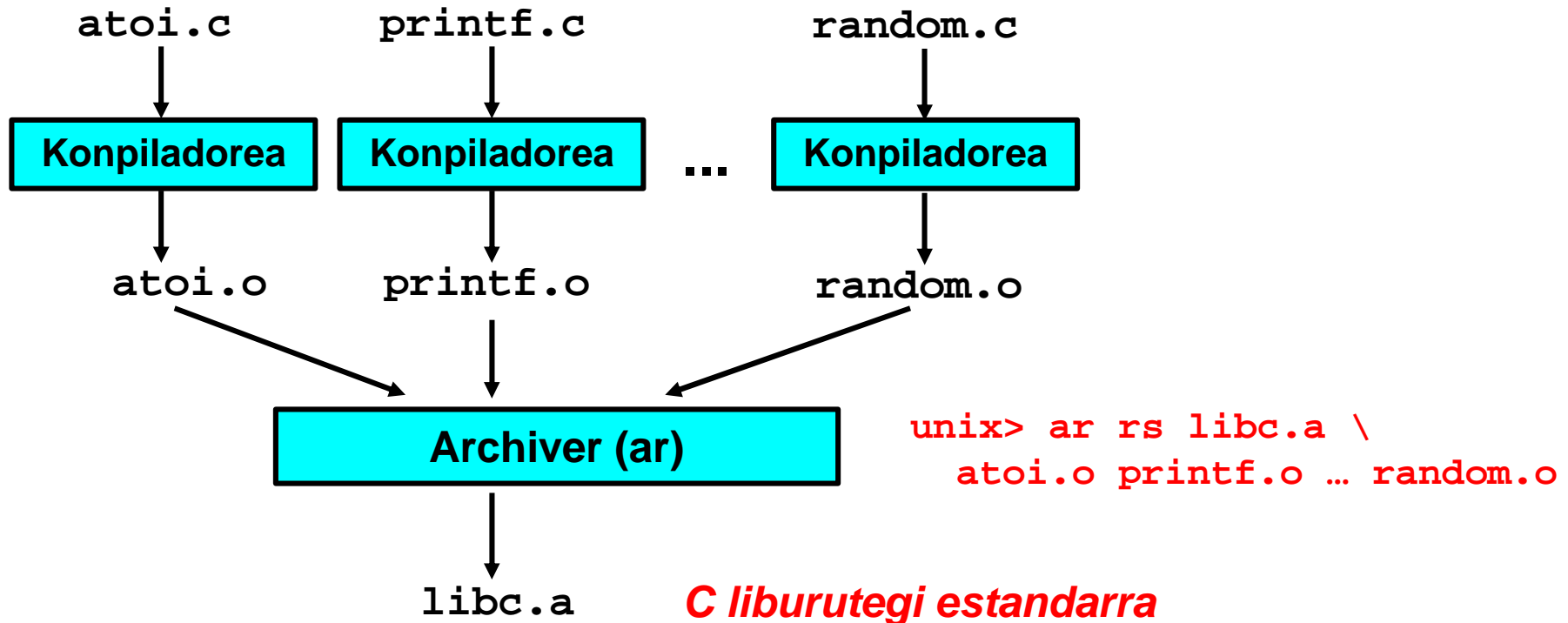
Estekatze estatikoa

- Liburutegia: elkar erlazionatutako objektu moduluen multzoa
- Sistemaren liburutegiak ala erabiltzaileak sortutakoak
- Liburutegi estatikoak:
 - Linkedizioa: programaren objektu moduluak eta liburutegiak estekatzea
 - Fitxategi exekutagarria: modulu guztiak barneratzen ditu
- Estekatze estatikoaren desabantailak:
 - Fitxategi exekutagarri handiak
 - Liburutegiko funtzioen kodea errepikatuta exekutagarri desberdinetan
 - Liburutegiko funtzioen kopia anitz memorian
 - Liburutegien eguneraketa: exekutagarri guztiak berriro linkatu behar

Estekatzeko estatikoa



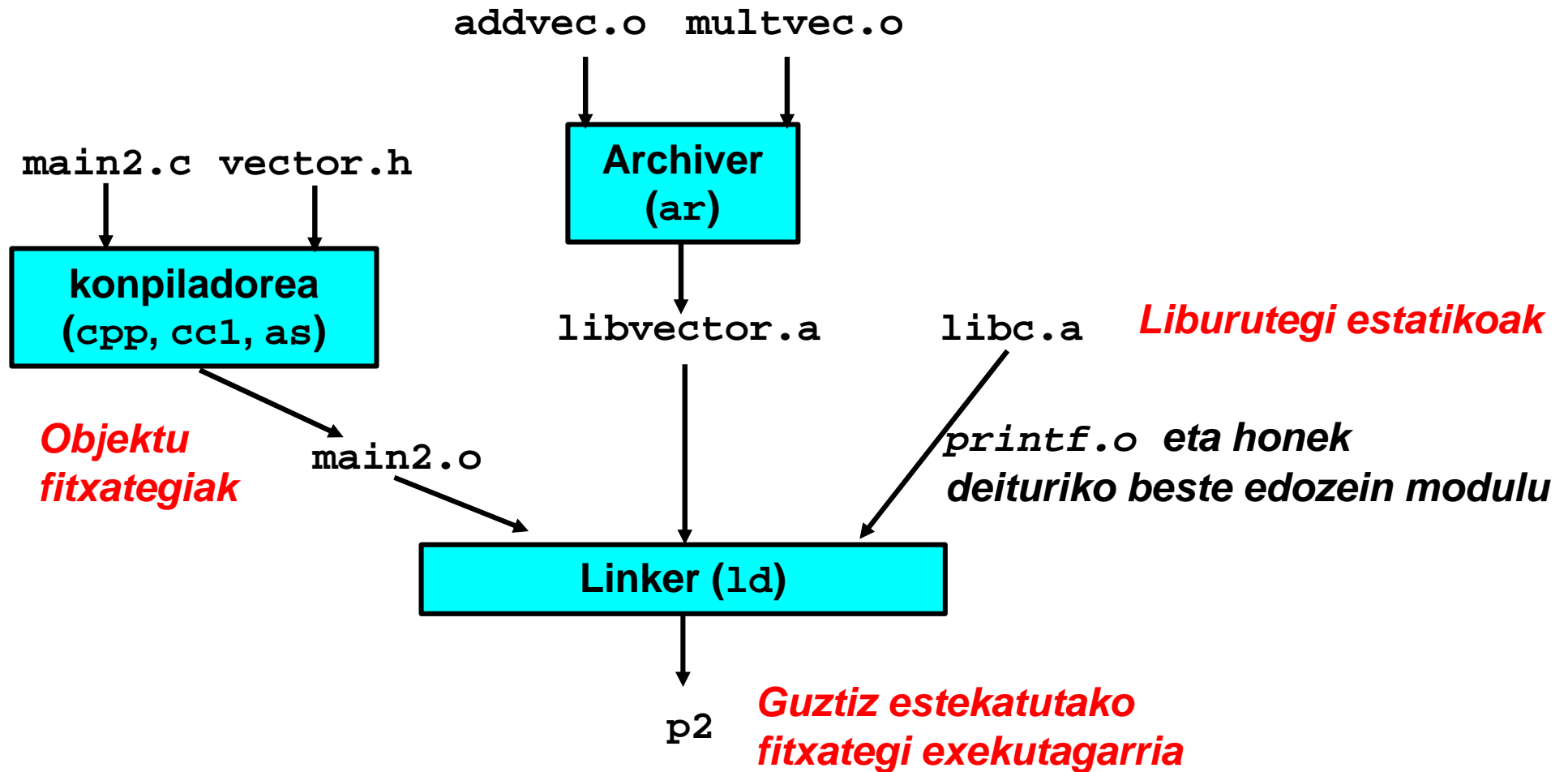
Liburutegi estatikoen sorrera



Archiver programak eguneraketa inkrementalak onartzen ditu:

- Birkonpilatuz liburutegian aldatu beharreko .o fitxategia

Estekatzea liburutegi estatikoekin



Estekatzeko dinamikoak

- Liburutegiaren karga eta muntaketa exekuzio unean
- Exekutagarriaren edukina:
 - Liburutegiaren izena
 - Programaren baitan exekuzio garaian kargatu eta estekatzeko errutina
 - Estekatzeko unekak:
 - Karga garaian
 - Liburutegiari lehenengo erreferentzia egiten zaionean
- Estekatzeko dinamikoaren abantailak:
 - Exekutagarriak tamaina txikiagokoak
 - Liburutegiko funtzioen kodea liburutegi fitxategian soilik
 - Prozesuek liburutegiaren kodea konpartitu dezakete
 - Liburutegien eguneraketa automatikoa: bertsioen erabilpena

Estekatzeko dinamiko (karga garaian)

