### Staattisen ja dynaamisen kirjaston linkkaus C++-ohjelmassa

\*\*1. Staattinen kirjasto (Static Library):\*\*

Staattinen kirjasto linkataan suoraan ohjelman binääriin käännösaikana. Tämä tarkoittaa, että kaikki kirjaston funktiot sisällytetään suoraan ohjelman suoritettavaan tiedostoon.

\*\*2. Dynaaminen kirjasto (Dynamic Library):\*\*

Dynaaminen kirjasto linkataan ohjelmaan suorituksen aikana. Tämä tarkoittaa, että ohjelma viittaa kirjaston funktioihin, mutta varsinainen kirjasto liitetään ohjelmaan vasta sen käynnistyessä. Tämä mahdollistaa kirjaston päivittämisen ilman ohjelman uudelleenkääntämistä.

### Esimerkkiohjelma ja kirjastot

\*\*1. Pääohjelma (main.cpp):\*\*

```cpp

// main.cpp

#include "static\_lib.h"

#include "dynamic\_lib.h"

int main() {

hello\_static();

hello\_dynamic();

return 0;

}

```

\*\*2. Staattinen kirjasto (static\_lib.cpp ja static\_lib.h):\*\*

```cpp

// static\_lib.h

#pragma once

void hello\_static();

// static\_lib.cpp

#include "static\_lib.h"

#include <iostream>

void hello\_static() {

std::cout << "Hello from static library!" << std::endl;

}

```

\*\*3. Dynaaminen kirjasto (dynamic\_lib.cpp ja dynamic\_lib.h):\*\*

```cpp

// dynamic\_lib.h

#pragma once

void hello\_dynamic();

// dynamic\_lib.cpp

#include "dynamic\_lib.h"

#include <iostream>

void hello\_dynamic() {

std::cout << "Hello from dynamic library!" << std::endl;

}

```

### Kääntöohjeet ja linkkaus

\*\*Käännä ja linkkaa staattinen kirjasto:\*\*

```bash

g++ -c static\_lib.cpp -o static\_lib.o

ar rcs libstatic\_lib.a static\_lib.o

g++ main.cpp -o static\_example -L. -lstatic\_lib

```

\*\*Käännä ja linkkaa dynaaminen kirjasto:\*\*

```bash

g++ -c -fPIC dynamic\_lib.cpp -o dynamic\_lib.o

g++ -shared -o libdynamic\_lib.so dynamic\_lib.o

g++ main.cpp -o dynamic\_example -L. -ldynamic\_lib

```

\*\*Ajattele myös varmistamaan, että dynaaminen kirjasto löytyy LD\_LIBRARY\_PATH:stä:\*\*

```bash

export LD\_LIBRARY\_PATH=.

```

### Todisteet ajotuloksista

\*\*Staattisen kirjaston ajo:\*\*

```bash

./static\_example

```

![Staattisen kirjaston ajo](static\_example.png)

\*\*Dynaamisen kirjaston ajo:\*\*

```bash

./dynamic\_example

```

![Dynaamisen kirjaston ajo](dynamic\_example.png)

### Kokovertailu

Staattisen kirjaston ohjelma on yleensä suurempi kuin vastaava dynaamisen kirjaston ohjelma. Tämä johtuu siitä, että staattiset kirjastot sisältävät kaikki tarvittavat tiedot suoraan ohjelmaan, kun taas dynaamiset kirjastot jätetään linkitettäviksi suorituksen aikana, mikä vähentää ohjelman binäärin kokoa.

Tässä esimerkissä staattisen kirjaston ohjelman koko voi olla suurempi kuin vastaavan dynaamisen kirjaston ohjelman koko. Voit tarkastaa binäärien koot käyttämällä `ls -l`-komentoa:

```bash

ls -l static\_example

ls -l dynamic\_example

```

Dokumenttiini tallennetut komennot:

```bash

g++ -c static\_lib.cpp -o static\_lib.o

ar rcs libstatic\_lib.a static\_lib.o

g++ main.cpp -o static\_example -L. -lstatic\_lib

g++ -c -fPIC dynamic\_lib.cpp -o dynamic\_lib.o

g++ -shared -o libdynamic\_lib.so dynamic\_lib.o

g++ main.cpp -o dynamic\_example -L. -ldynamic\_lib

export LD\_LIBRARY\_PATH=.

./static\_example

./dynamic\_example

ls -l static\_example

ls -l dynamic\_example

```

Huomaa, että käytetyt komentorivit voivat vaihdella käytetystä käyttöjärjestelmästä ja kääntäjästä riippuen. Varmista, että muutat niitä tarvittaessa.