

COMPUTER ORGANIZATION AND DESIGN

SUBSUBSECTION

LINKER@2.12

Dynamically Linked Library:

1-1:C言語におけるビルド階層

Cソースコード (test.c)

↓コンパイラ

アセンブリコード (test.s)

↓アセンブラ

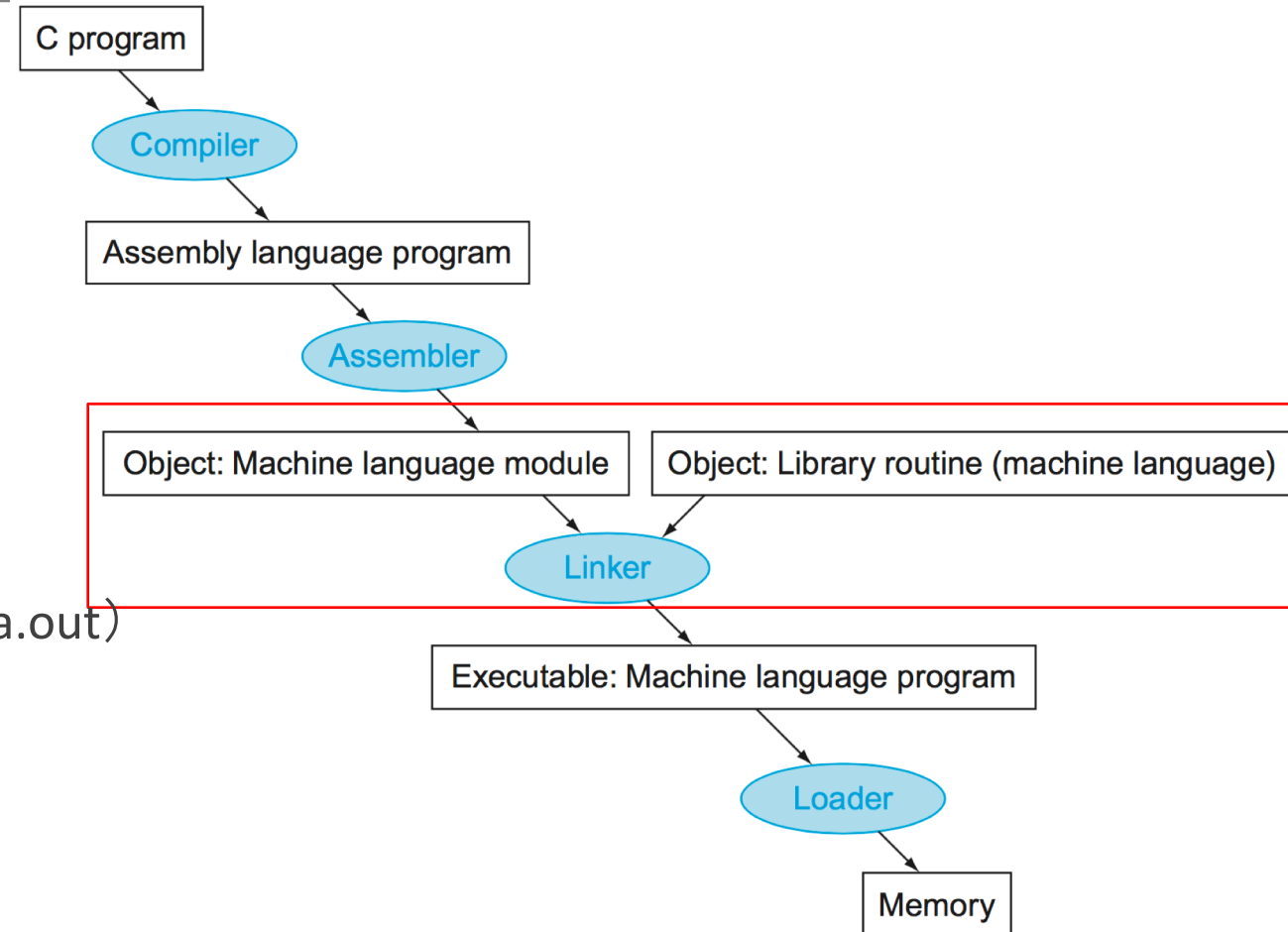
オブジェクトモジュール:参照未解決 (test.o)

↓リンカ

オブジェクトモジュール:参照解決:実行可能 (a.out)

↓ローダ

メモリ配置(実行)



Dynamically Linked Library:

1-2: リンカ

リンカ導入前:

- あるモジュールのソースコード1行を変更
- 全体コンパイルと全体アセンブルし直し

 : machine lang(compiled)

 : object(assembled)

-
-
-
-
-
-
-
-
-



Dynamically Linked Library:

1-2: リンカ

リンカ導入前:




- あるモジュールのソースコード1行を変更
- 全体コンパイルと全体アセンブルし直し

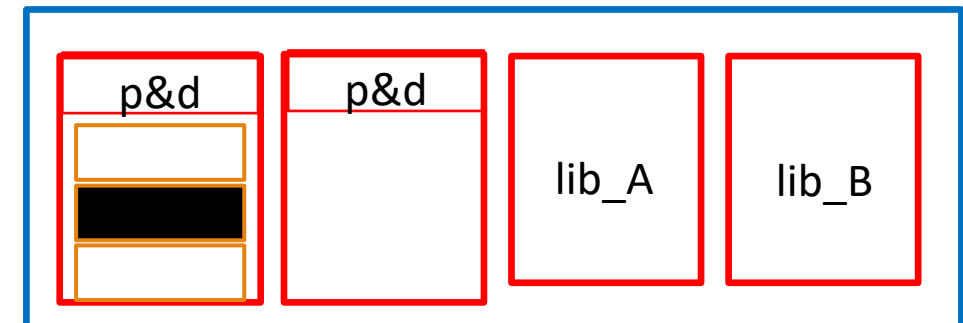
リンカ導入後:

- 変更のないモジュール(特にライブラリなど)は
- コンパイル、アセンブルが不要化
- ビルドを高速化

リンカは外部参照となったルーチン同士を繋ぐ

-
-
-
-
-

 : machine lang(compiled)
 : object(assembled)
 : object(linked)

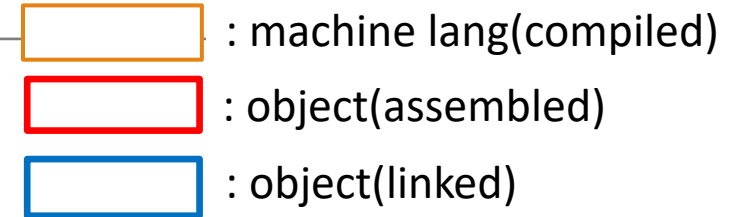


Dynamically Linked Library:

1-2: リンカ

リンカ導入前:

- あるモジュールのソースコード1行を変更
 - 全体コンパイルと全体アセンブルし直し



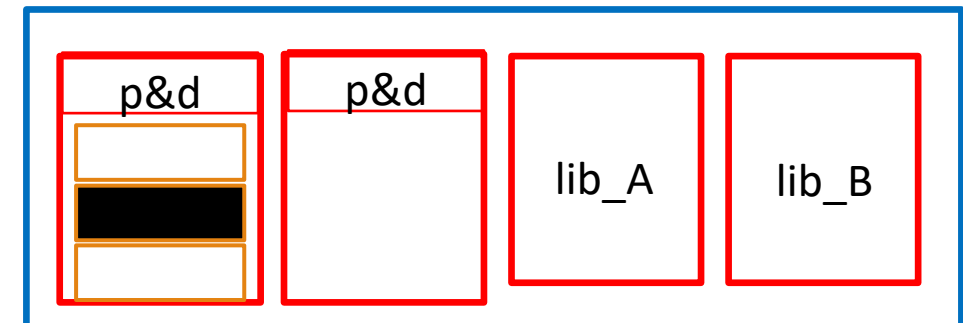
リンカ導入後:

- 変更のないモジュール(特にライブラリなど)は
- コンパイル、アセンブルが不要化
 - ビルドを高速化

リンカは外部参照となったルーチン同士を繋ぐ

静的リンカ:

- プログラム実行前にライブラリとのリンクを行う
- 欠点1:ライブラリ自体に変更を反映できない
 - 古いバージョンを使い続ける
- 欠点2:使わないルーチンもリンクされる
- 動的リンクライブラリ(DLL)の発想へ



Dynamically Linked Library: 1-3: DLL

プログラム実行時にライブラリをリンク/ロード

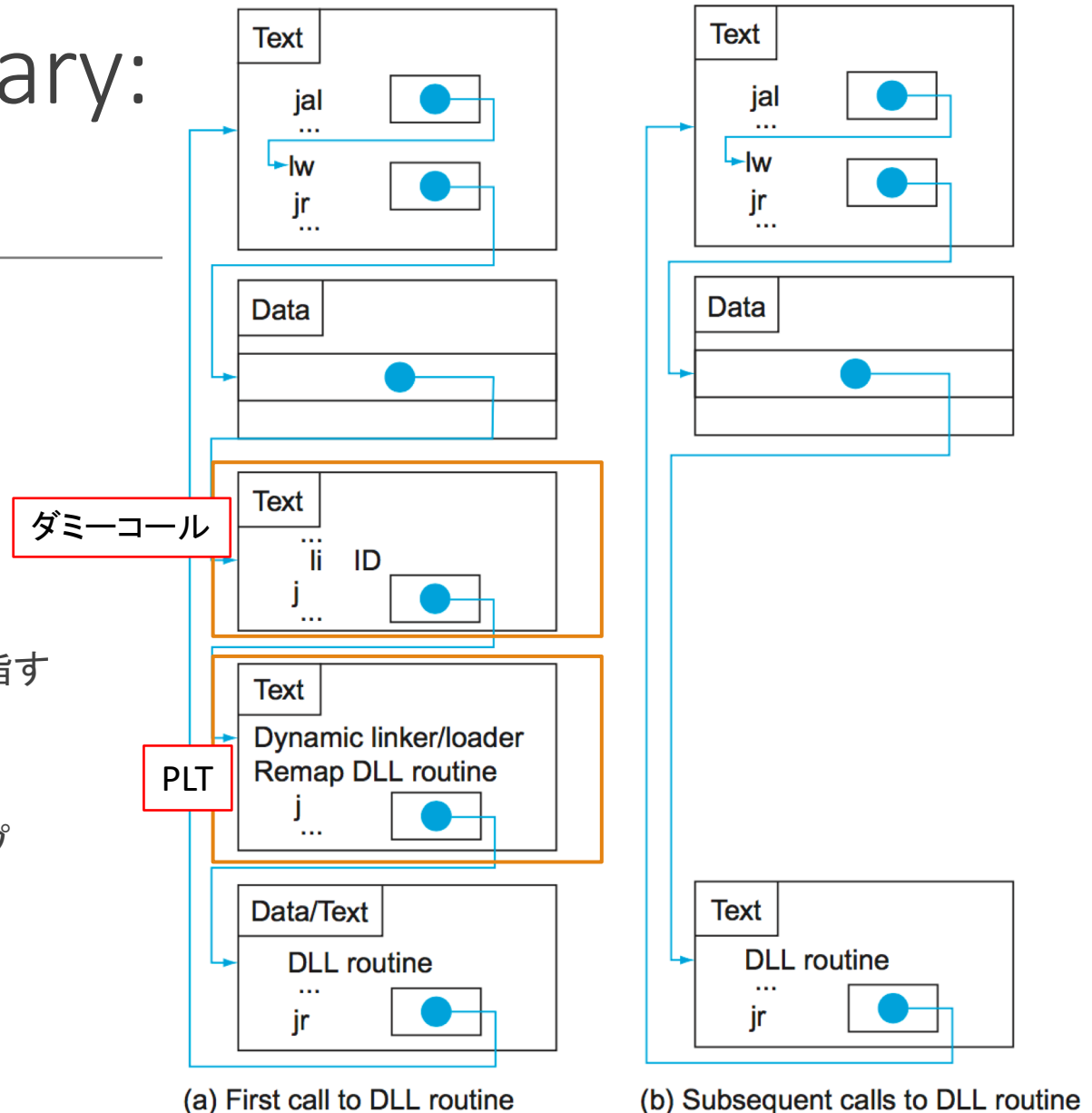
- 呼び出されたライブラリが適宜リンク、リマップ
 - 必要ないライブラリはリンクされない
 - ライブラリの更新も反映

Lazy Procedure Linkage:

- 上記の必要な部分のみリンクする動作
 - ルーチン呼び出しは関節参照によってダミーコールを指す
 - DCとPLTによってルーチンを再配置
- Procedure Linkage Table(PLT):**
 - DCを指していた間接参照を呼び出しルーチンにリマップ
 - 正確にはPLTに載っている対応関係を元にリマップ

オーバーヘッド:

- PLTなどのテーブルの空間オーバーヘッド
- 初回再配置時の実行オーバーヘッド



参考文献

Computer Organization and Design RISC-V EDITION

- David A. Patterson, John L. Hennessy

例解UNIXプログラミング教室 システムコールを使いこなすための12講

- 権藤克彦, 富永和人