

システムプログラムレポート

2024年11月26日
学籍番号：35714121
名前：福富隆大

1. 6-4ページの例6.3(a)のe6-1.cと6-5ページのe6-1m.cをgccでコンパイルし、実行ファイルe6-1.exeを作成しなさい

作成したファイル内容、結果について

```
gcc e6-1.c e6-1m.c -o e6-1.exe
```

というコマンドを実行してe6-1.cとe6-1m.cをgccでコンパイルし、実行ファイルe6-1.exeを作成した

2. 6-12ページのコマンド実行例を参考にして、nmとobjdumpを実行し、変数sendai, yokohama, nagoyaを確認しなさい。そして、6-12ページ（arm）の結果と比較・考察しなさい

実行結果

nm実行結果

```
00000000100008000 0000000000000000 D _sendai
0000000010000800c 0000000000000000 D _yokohama
00000000100008018 0000000000000000 D _nagoya
```

6-12ページ（arm）のnm結果

```
0002102c 0000000c D sendai
00021038 0000000c D yokohama
00021044 0000000c D nagoya
```

objdump実行結果

```
e6-1.exe:          file format mach-o arm64
Contents of section __DATA,__data:
 100008000 00000000 02000000 02000000 00000000 .....
 100008010 04000000 05000000 00000000 05000000 .....
 100008020 02000000                ....
```

6-12ページ (arm) のobjdump結果

```
> arm-objdump e6-1a-arm.exe -s --start=0x2102c --stop=0x21050
...
2102c 00000000 02000000 02000000 00000000 .....
2103c 04000000 05000000 00000000 05000000 .....
2104c 02000000 ....
```

作成したファイル内容、結果について

nmコマンドを実行するとまず、サイズが違っていた。これは、実行したパソコンが64bitだからだと予想する。どちらもDがついているのでデータセクションに存在していることがわかる。配列のサイズはスライドの結果と変わらず12バイトだった。

objdumpコマンドを実行してみるとスライドの結果と同じ値が入っていた。

3. 6-13ページのコマンド実行例を参考にして、最適化したコードと最適化しないコードを作成し、変数sendai,yokohama,nagoyaを比較しなさい。

実行結果

最適化したコードのnm実行結果

```
00000000100008000 0000000000000000 D _sendai
0000000010000800c 0000000000000000 D _yokohama
00000000100008018 0000000000000000 D _nagoya
```

最適化していないコードのnm実行結果

```
00000000100008000 0000000000000000 D _sendai
0000000010000800c 0000000000000000 D _yokohama
00000000100008018 0000000000000000 D _nagoya
```

作成したファイル内容、結果について

今回は-Oを用いて最適化しても特に変わらなかった。これは3つの配列がコードサイズや実行効率などに影響がなかったか、少なかったからだと予想する。また、コンパイラのバージョンによってもかわるのではないかと思う。

講義に対する感想・質問・意見

最初の頃はコマンドを見てもどこの部分が何を指定しているのかわからず、実行結果についてもなかなか理解できないことが多かったが、スライドのコマンドと自分の環境でのコマンドがなぜちがっていて、どうすればうまくいくのかがだんだん分かってきた。アドレスを扱うことにもだんだん慣れてきて楽しい。