

1 使用ファイル

2 作業 1: アセンブラへの変換

使用ファイルは以下の通りである。

```
main.c
mk1cal.s
monthwoffset.s
monthlen.s
divsub.s
modsub.s
```

以下の図 2.1 にコールグラフを示す。

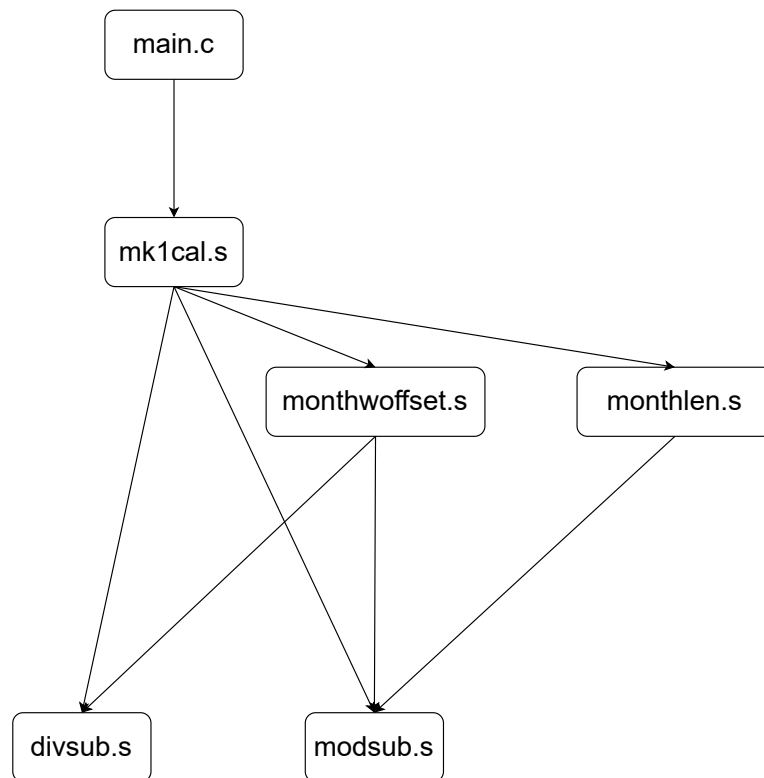


図 2.1: 作業 1 のコールグラフ

2.1 作業 1 のプログラム

以下に作業 1 で使用するプログラムを解説する。

`main.c` はカレンダーを表示するプログラムで、I/O を行う。`IW,CW,OW,OH,MAX_CANVAS` はカレンダーの表示に使用されるいくつかの定数である。それぞれ、1 週間の長さ、1 日を出力するためのマス目幅、はカレンダーの横幅（週の日数）、カレンダーの縦幅（週の行数）、カレンダーの描画に使用される `char` を保持する文字列を表す。`printcal` 関数は、`canvas` の内容を表示するための関数である。`pch`, `pdecweek`, `week` 関数は、特定の文字や週の情報を表示するために使用している。

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 extern int monthwoffset(int, int, int);
6 extern int monthlen(int, int);
7 extern int mk1cal(int, int, char *);
8
9
10 #define IW (7)
11 #define CW (3)
12 #define OW (21)
13 #define OH (6)
14 #define MAX_CANVAS (OW*OH)
15
16 char canvas[MAX_CANVAS];
17
18 void
19 printcal(void)
20 {
21     int i, j;
22     for(i=0;i<MAX_CANVAS;i++) {
23         if(canvas[i]) {
24             printf("%c", canvas[i]);
25         }
26         if(i%OW==OW-1) {
27             printf("\n");
28         }
29     }
30 }
31
32 int main(int argc, char *argv[])
33 {
34     int y; //year
35     int m; //month
36     if(argc==3) {
37         y = atoi(argv[1]);
38         m = atoi(argv[2]);
39     }
40     else {
41         printf("usage: %s y m\n",
42             argv[0]);
43         exit(9);
44     }
45
46     memset(canvas, '\0', sizeof(canvas));
47     mk1cal(y, m, canvas);
48     printcal();
49     exit(0);
50 }

```

図 2.2: 作業 1 の main.c

以下の divsub.s はアセンブリは割り算をするときに使用される。引数は r0 と r1 であり、r0/r1 の結果を r0 で返す。除算は引き算をループして実装しており、その条件は、「cmp r0, r1」で、r0 が r1 より小さいときにループを抜ける。その間、ループ内では r2 にループ回数がインクリメントされる。最後に r2 を r0 に代入している。

```

1      .section .text
2      .global divsub
3      .type divsub, %function
4      @r0/r1
5      divsub:
6          push {r2,lr}
7          mov r2, #0
8      loop:
9          cmp r0,r1
10         blt loopout
11         add r2,r2,#1
12         sub r0,r0,r1
13         b loop
14      loopout:
15         mov r0,r2
16         pop {r2,lr}
17         bx lr

```

図 2.3: 作業 1 の divsub.s

以下の modsub.s では、r0/r1 の剰余を r0 として返している。除算は引き算をループして実装しており、その条件は、「cmp r0, r1」で r0 が r1 より小さいときにループを抜ける。引けなくなった r0 が剰余ということである。

```

1      .section .text
2      .global modsub
3      .type modsub, %function
4      modsub:
5      loop:
6      @Input r0,r1
7      @r1 == #7
8          cmp r0, r1
9          blt loopout
10         sub r0, r0, r1
11         b loop
12      loopout:
13         bx lr

```

図 2.4: 作業 1 の modsub.s

mk1cal.s、monthlen.s、monthwoffset.s は行数が多く、1 ページに入り切らなかったので載せないが、zip ファイルに同封している。

以下に説明を述べる。

monthwoffset.s はツェラーの公式に則して入力された西暦の月の曜日の出力を行っている。monthlen.s は、閏年も加味した入力された西暦の月の月の長さの出力を行っている。mk1cal.s は、上記の情報から、日曜日始まりにするように char 型の canvas に数列と空はくを代入することによって、カレンダーの出力情報を作成している。

2.2 作業 1 の出力結果

以下の*に作業 1 の出力結果を示す。

```

m20046@hard2-rpi1:~/calendar/test/a $ cc divsub.s modsub.s mk1cal.s monthlen.s
monthwoffset.s main.c
m20046@hard2-rpi1:~/calendar/test/a $ ./a.out 2023 11

    1  2  3  4
  5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30

```

図 2.5: 作業 1 の出力結果

3 作業 2: アセンブラプログラムの拡張

3.1 (a)

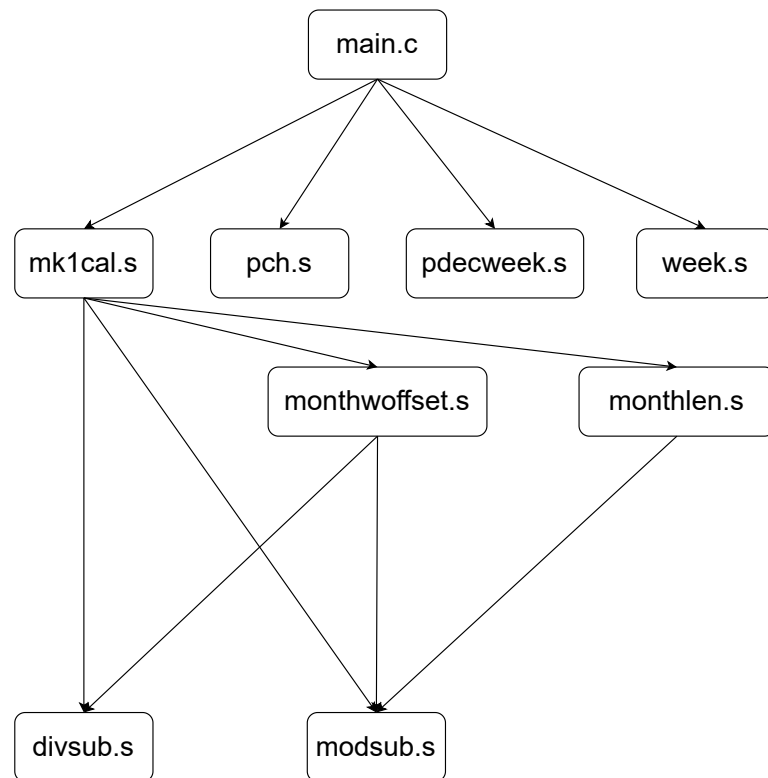


図 3.1: 作業 2(a) のコールグラフ

3.1.1 作業 2(a) のプログラム

作業 2(b) のプログラムにおいて、main.c、divsub.s、modsub.s、mk1cal.s、monthlen.s、monthwoffset.s は作業 1 と同じである。pdecweek.s は年号を出力するために使用している。pch は引数の char 型の文字列を出力するために使用している。今回は '/' と改行を出力するために使用した。week は ASCII で指定した曜日を出力するために使用した。

3.1.2 作業 2(a) の出力結果

```
m20046@hard2-rpi1:~/calendar/test $ cc divsub.s modsub.s mk1cal.s
monthlen.s monthwoffset.s main.c pch.s pdecweek.s week.s
m20046@hard2-rpi1:~/calendar/test $ ./a.out 2023 11
    2023/11
Su Mo Tu We Th Fr Sa
      1  2  3  4
  5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30
```

図 3.2: 作業 2(a) の出力結果

3.2 (b)

以下にコールグラフを示す。

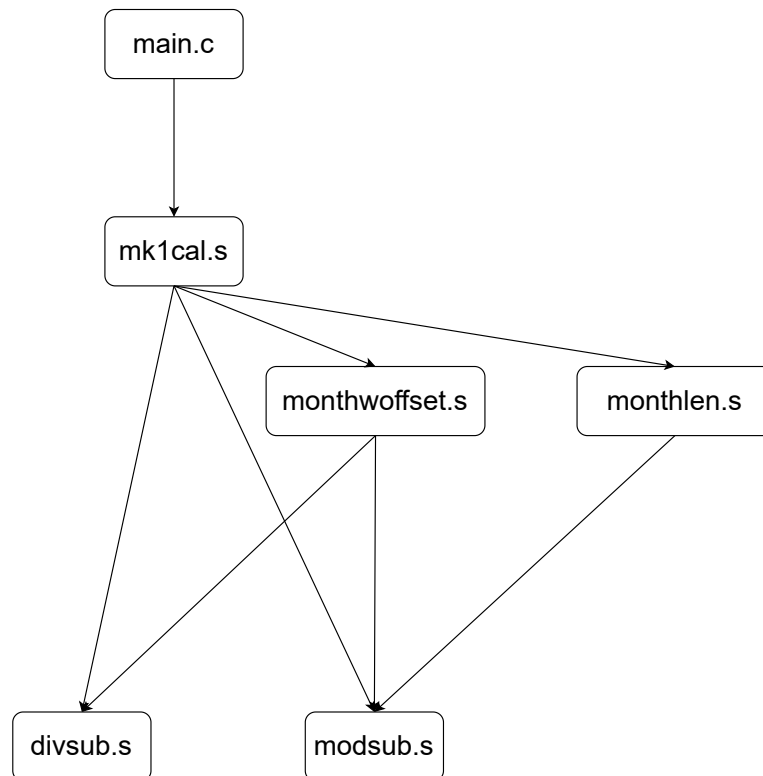


図 3.3: 作業 2(b) のコールグラフ

3.2.1 作業 2(b) のプログラム

作業 2(b) のプログラムにおいて、`main.c`、`divsub.s`、`modsub.s`、`mk1cal.s`、`monthlen.s` は作業 1 と同じである。`monthwoffset.s` は、7 での剰余を求める前に 6 を足して、月曜日始まりになるようにずらしている。

3.2.2 作業 2(b) の出力結果

```
m20046@hard2-rpi1:~/calendar/test/b $ cc divsub.s modsub.s mk1cal.s
monthlen.s monthwoffset.s main.c
m20046@hard2-rpi1:~/calendar/test/b $ ./a.out 2023 11
    1  2  3  4  5
  6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30
```

図 3.4: 作業 2(b) の出力結果

3.3 (c)

以下の図 2.1 にコールグラフを示す。

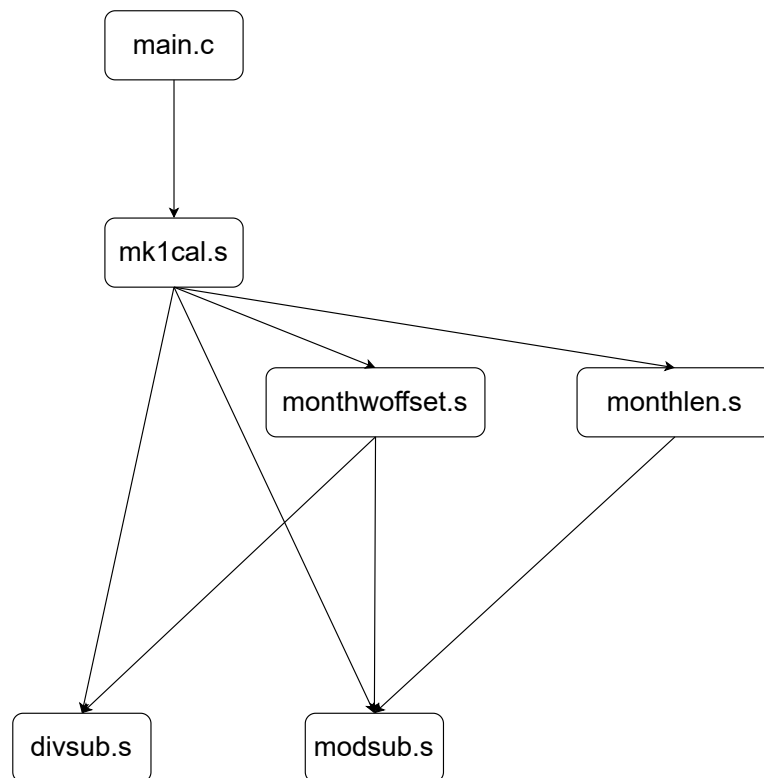


図 3.5: 作業 2(c) のコールグラフ

3.3.1 作業 2(c) のプログラム

作業 2(c) のプログラムにおいて、divsub.s、modsub.s、monthlen.s、monthwoffset.s は作業 1 と同じである。mk1cal.s は、mkall.loop: のところで、cmp を使用し、3ヶ月分ループしており、 $b = r * 66 + c * 3 + (\text{ループ回数}) * 22$ としている。66 は、横幅の 3 ヶ月とスペースを考えると、一週間分ずらしている。c*3 は 1 マス分ずらしており、(ループ回数)*22 は、横に一週間分シフトしている。main.c は、canvas の大きさを 3 倍にした。

3.3.2 作業 2(c) の出力結果

```
m20046@hard2-rpi1:~/calendar/threemonth/test $ cc divsub.s mk1cal.s
modsub.s monthlen.s monthwoffset.s main.c
m20046@hard2-rpi1:~/calendar/threemonth/test $ ./a.out 2023 8
      1  2  3  4  5              1  2  1  2  3  4  5  6  7
  6  7  8  9 10 11 12    3  4  5  6  7  8  9    8  9 10 11 12 13 14
13 14 15 16 17 18 19    10 11 12 13 14 15 16    15 16 17 18 19 20 21
20 21 22 23 24 25 26    17 18 19 20 21 22 23    22 23 24 25 26 27 28
27 28 29 30 31          24 25 26 27 28 29 30    29 30 31
```

図 3.6: 作業 2(c) の出力結果

3.4 (d)

以下にコールグラフを示す。

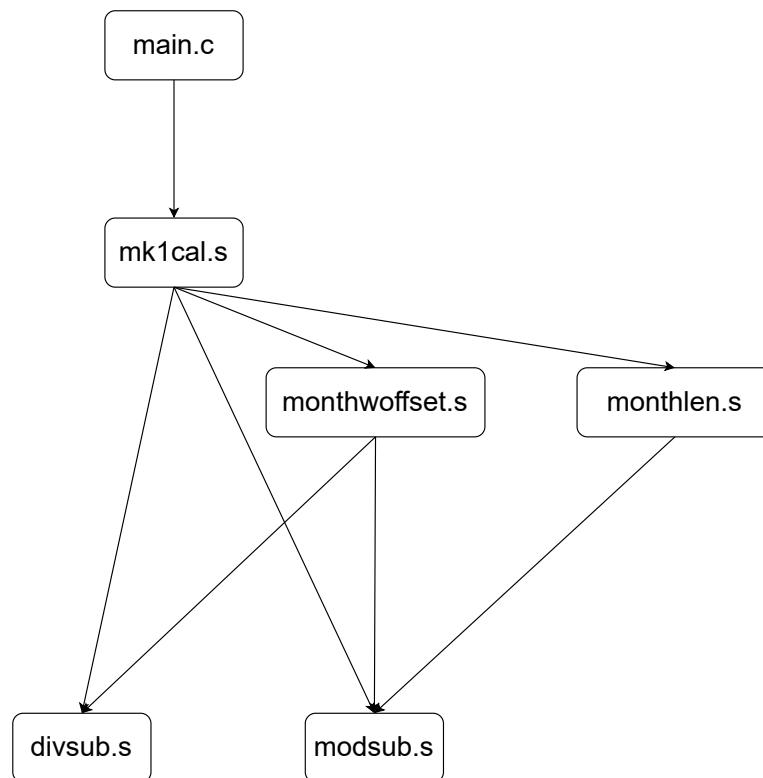


図 3.7: 作業 2(d) のコールグラフ

3.4.1 作業 2(d) のプログラム

作業 2(b) のプログラムにおいて、`main.c`、`divsub.s`、`modsub.s`、`monthwoffset.s`、`monthlen.s` は作業 1 と同じである。`mk1cal.s` を変更している。第 1 引数と第 3 引数である西暦*12、第 2 引数と第 4 引数である月を引き算してその分ループを回して出力している。

3.4.2 作業 2(d) の出力結果

```
m20046@hard2-rpi1:~/calendar/range $ cc divsub.s modsub.s mk1cal.s
monthwoffset.s monthlen.s main.c
m20046@hard2-rpi1:~/calendar/range $ ./a.out 2023 1 2023 3
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

      1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28

      1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31
```

図 3.8: 作業 2(d) の出力結果