

基礎ソフト実験 UNIX・アセンブラ実習 第6回

課題 下記のプログラムは、256バイトの大きさを持つキューを作り、それに1バイトずつデータを書き込むプログラムである。このプログラムについて、以下の設問に答えよ。

- (1) プログラムのフローチャートを作成せよ。
- (2) プログラム中の(あ)～(う)の箇所の処理内容(意味)を説明せよ。
- (3) プログラム中の(え)には、レジスタ復帰の処理が必要である。完成させよ。
- (4) このプログラムを emacs で入力し、ステップ実行で動作を確認せよ。
- (5) データの書き込みが256回目(In_Q呼び出す直前で、d4の値は1)の時、PUT_BUF内の主要なレジスタとPUT_PTR,GET_PTR,PUT_FLG,GET_FLGの変化を表にして確認せよ(詳細は出席票参照)。
- (6) データの書き込みが257回目の時、PUT_BUF終了後、PUT_FLGの値はいくらになっているか。また、なぜその値になったのかを説明せよ。

```
1 .section .text
*****
** メインルーチン
** a0:書き込むデータのアドレス
** d4:書き込み回数
*****
2 Start:
3     jsr    Init_Q /* キューの初期化処理 */
4     lea.l   Data_to_Queue, %a0
5     move.l  #257, %d4
6 Loop1:
7     subq.w  #1, %d4
8     bcs     End_program
9     jsr     In_Q /* 書き込み処理 */
10    bra     Loop1
11 End_program:
12    stop    #0x2700
*****
** キューの初期化処理
*****
13 Init_Q:
14    lea.l   BF_START, %a2
15    move.l  %a2, PUT_PTR
16    move.l  %a2, GET_PTR
```

```

17      move.b  #0xff, PUT_FLG
18      move.b  #0x00, GET_FLG
19      rts


*****

** In_Q キューへのデータ書き込み
** a0:      書き込むデータのアドレス
** d0:      結果 (00:失敗, 00以外:成功)
*****

20 In_Q:
21      jsr PUT_BUF /* キューへの書き込み */
22      rts

*****

** PUT_BUF
** a0:      書き込むデータのアドレス
** d0:      結果 (00:失敗, 00以外:成功)
*****

23 PUT_BUF:
24      movem.l  %a1-%a3, -(%sp)
25      move.b   PUT_FLG, %d0
26      cmp.b    #0x00, %d0
27      beq      PUT_BUF_Finish
28      movea.l  PUT_PTR, %a1
29      move.b   (%a0)+, (%a1)+
30      lea.l    BF_END, %a3
31      cmpa.l   %a3, %a1
32      bls      PUT_BUF_STEP1
33      lea.l    BF_START, %a2
34      movea.l  %a2, %a1
35 PUT_BUF_STEP1:
36      move.l   %a1, PUT_PTR
37      cmpa.l   GET_PTR, %a1
38      bne      PUT_BUF_STEP2
39      move.b   #0x00, PUT_FLG
40 PUT_BUF_STEP2:
41      move.b   #0xff, GET_FLG
42 PUT_BUF_Finish:
43      
44      rts

```

(あ)

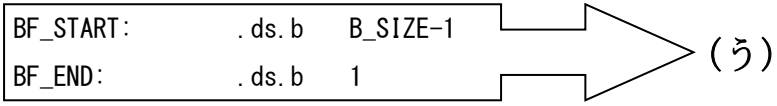
(い)

(え)

```

45 .section .data
*****
** キュー用のメモリ領域確保
*****
46         .equ    B_SIZE, 256
47 BF_START:    .ds. b    B_SIZE-1
48 BF_END:      .ds. b    1
49 PUT_PTR:     .ds. l    1
50 GET_PTR:     .ds. l    1
51 PUT_FLG:     .ds. b    1
52 GET_FLG:     .ds. b    1
*****
** 書き込むデータ (サンプル)
*****
53 Data_to_Queue: .ascii  "ABC"
54 .end

```



(う)

今日の実習はここで終了です.