

計算機システムレポート 第4回

05D37024 實本英之

平成 18 年 8 月 9 日

1 課題 1:例外ハンドラ

プログラムは以下のファイル
ex4-1.s テスト用の実行ファイル
myexception.s 実装した例外ハンドラ

1.1 解説

大まかな流れは、以下のようになっている。各項目のカッコ内の数字は該当行

1. \$t0 用の保存場所定義 (2-5)
2. bne を使うための \$at 退避 (7-10)
3. \$t0 の退避 (11-12)
4. 例外番号の取得 (14-17)
5. 例外番号が 13 であるかチェック、適合しなければ 8 へ (18-20)
6. syscall 番号が 100 であるかチェック、適合しなければ 8 へ (22-24)
7. 入力値を 10 倍する処理 (25-27)
8. EPC を 4 進める (28-32)
9. \$t0 の復帰 (34-35)
10. \$at の復帰 (36-39)
11. 例外終了の eret (41-42)

さらに、main ラベルを呼び出すためにスライドに書かれていたコードを追加してある (43-50)

1.2 実行結果

```
Input number? : 4
40
```

2 課題 2:caller-save と callee-save の違い

プログラムは以下のファイル
ex4-2.s callee-save での実装

2.1 解説と回答

callee-save ではサブルーチン呼び出し前後の `sw`, `lw` を行う代わりにサブルーチン内でレジスタ退避を行う。read1bit では `$v0` 以外のレジスタを使用していないため、callee-save 時に退避が必要なレジスタはない。

以下回答。

caller-save 時、このプログラムは read1bit 前と後で `$t0`, `$t1` が同じ値でなければいけないため、`$ra` 以外にこの 2 つのレジスタ (2word) を退避する必要がある。一方、callee-save 時は、read1bit が `$v0` 以外のレジスタを使用していない。`$v0` は戻り値を入れておくレジスタであるので値を退避 / 復元する必要がない。よって callee-save 時は退避 / 復元の必要があるレジスタは `$ra` のみ。以上より、このプログラムにおいては callee-save の方が効率がよい。

3 感想

(あっても無くてもよいが、あったほうが TA がうれしい。得点にはあまり影響しない。)
今回は第 4 回目のレポートを例に取りましたが、実際は 2,3,4 回目のプログラム解説はコメント以上のことはしなくていいと言ったので、プログラム内に直接コメントを書き込んでしまっても OK です。ただし 4-2 については必ず別紙に書いてください。