

HW5

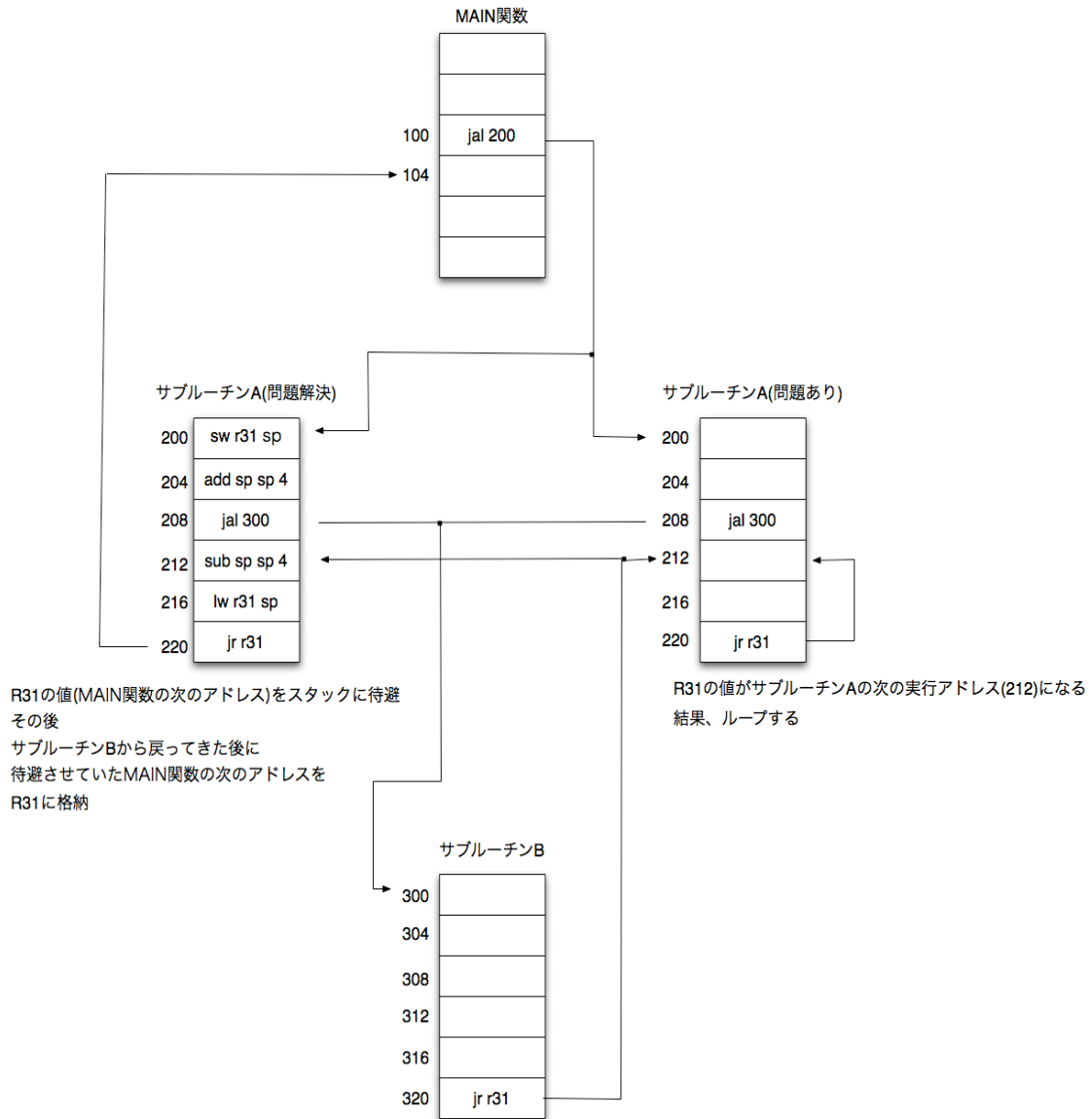
学籍番号 :e075739A

名前 :津波古正輝

日付 :12月4日(木)

クイズ 1

MAIN 関数から関数 A を呼び出し、関数 A から関数 B を 2 重に呼び出しする場合には、教科書のままでは不具合が発生する。2 重呼び出し時の表 3F 相当のプログラムを作成し、そのプログラムをどう修正すれば良いか？



図：不都合な場合(右)とその解決策(左)

2 重呼び出しを行う前に、R31(次に実行すべきアドレスが格納されている)を待避させなければならない。

クイズ 2

CALLEE SAVE での表 3F を作成せよ。CALLER SAVE と CALLEE SAVE のメリットでメリットを述べよ。

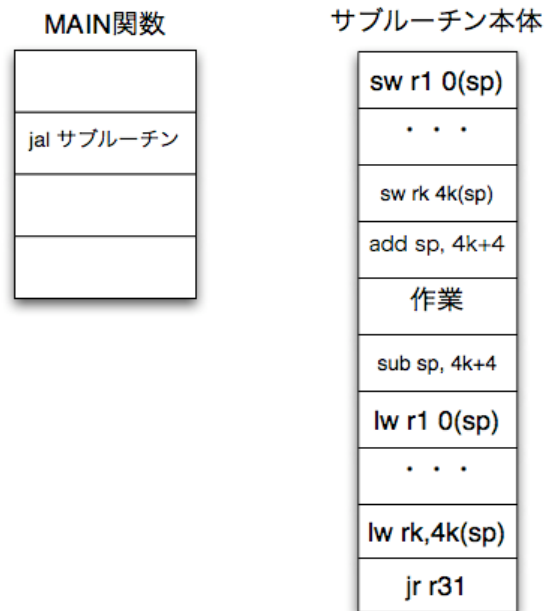


図 : CALLEE SAVE バージョンの 3.F

CALLERSAVE

メリット

デメリット

- ・ サブルーチンを呼び出すたびにレジスタの値の待避と復帰を行わないといけなないので、MAIN 関数が長くなる。

CALLEESAVE

メリット

- ・ MAIN 関数が短くなる。
- ・ PUSH と POP のアクセス時間が短くなる。

デメリット

- ・ サブルーチンが長くなる(サブルーチンは短いほうがよい)。

クイズ 3

3.D の 1 と 6 の手順を短縮する方法を提案せよ。

3.D の手順はコーラセーブ方式である。コーラセーブ方式(MAIN プログラムでスタックに待避)を行う場合、MAIN プログラムとスタック領域とに間があるため、アクセス時間に多少の時間がかかってしまう。コーリセーブ方式(サブルーチンでスタックに待避)を行うことで、アクセス時間を短縮することができる。つまり、3.D のコーラセーブ方式をコーリセーブ方式に変更すればよい。(この方法は、スタック領域が MAIN プログラムから離れた位置に領域を確保されていると仮定する。また、サブルーチンの記述が MAIN プログラムよりスタック領域に近いと仮定する。)

3.D サブルーチンの手順(コーリセーブ方式)

- 1:戻り値の待避
- 2:サブルーチンの先頭アドレスへのジャンプ
- 3:レジスタ値の待避
- 4:サブルーチン本体の実行
- 5:レジスタ値の復帰
- 6:戻り番地へのジャンプ
- 7:もとの命令列の実行再開

図：コーリセーブ方式による 3.D