
基礎から学ぶ電子計算機工学

Advanced Study 2

修正リスト

2022/9/18 作成

修正リスト

項	スライド番号	修正内容	修正日
①	203	sw疑似命令の表記など	2022/8/31
②	156	アルゴリズム修正	2022/9/18

①

疑似命令(ロード, ストア)

疑似命令	オリジナルの命令	コメント
<i>la rd, symbol</i>	<i>auipc rd, symbol[31:12]</i> <i>addi rd, rd, symbol[11:0]</i>	Load address
<i>lw rd, symbol</i>	<i>auipc rd, symbol[31:12]</i> <i>lw rd, symbol[11:0](rd)</i>	Load global
<i>sw rs, symbol, tmp</i>	<i>auipc tmp, symbol[31:12]</i> <i>sw rs, symbol[11:0](tmp)</i>	Store global
<i>li rd, imm</i>	<i>lui rd, imm[31:12]</i> <i>addi rd, rd, imm[11:0]</i>	Load 32bit immediate

Load globalはSymbolラベルの付いたデータをレジスタにロードする疑似命令
lwだけでなく、lh, lbでも使用可能である。

Store globalはSymbolラベルの付いたデータをレジスタからメモリにストアする疑似命令
swだけでなく、sh, sbでも使用可能である。

注: *rs, rd*は任意のレジスタ名, *tmp*はzeroレジスタ以外のレジスタ名, *symbol*はデータラベル, *imm*は 整数(最大32bit)

②

文字列の扱い

- 文字列配列の大きさは使用する文字列の長さよりも十分大きい大きさの配列を準備しておく。実際の文字列の長さは配列の大きさより短くても構わない。
- 文字列の最終文字がNull文字であることを利用して文字列の処理を終了するようにプログラムを書く必要がある。文字列の長さはわからなくてよい。

文字列” AOYAMA”を大文字の” aoyama”に変換するプログラムの例

```
// 大文字→小文字変換
char s[256], p[256];
s="AOYAMA";
while(*s != NULL) {
    if((*s >= 0x41) && (*s <= 0x5A)) {
        *p = *s + 0x20;
    } else {
        *p = *s;
    }
    p++;
    s++;
}
*p = NULL; // add NULL character
```

← 256文字まで格納できる文字列用の配列を宣言する。

← sは6文字+NULL文字の文字列の先頭を示すポインタ

← 文字列の終了判定

← 英字大文字は0x41以上0x5A以下であることで判定

← 文字列のポインタを一つ進める。

← 文字列の最後にNULL文字を追加