

計算機構成論

Lecture 10 まとめ

2023年度前期
情報理工学部 Rクラス担当
越智裕之

「計算機構成論」 (2) 他の授業との関係

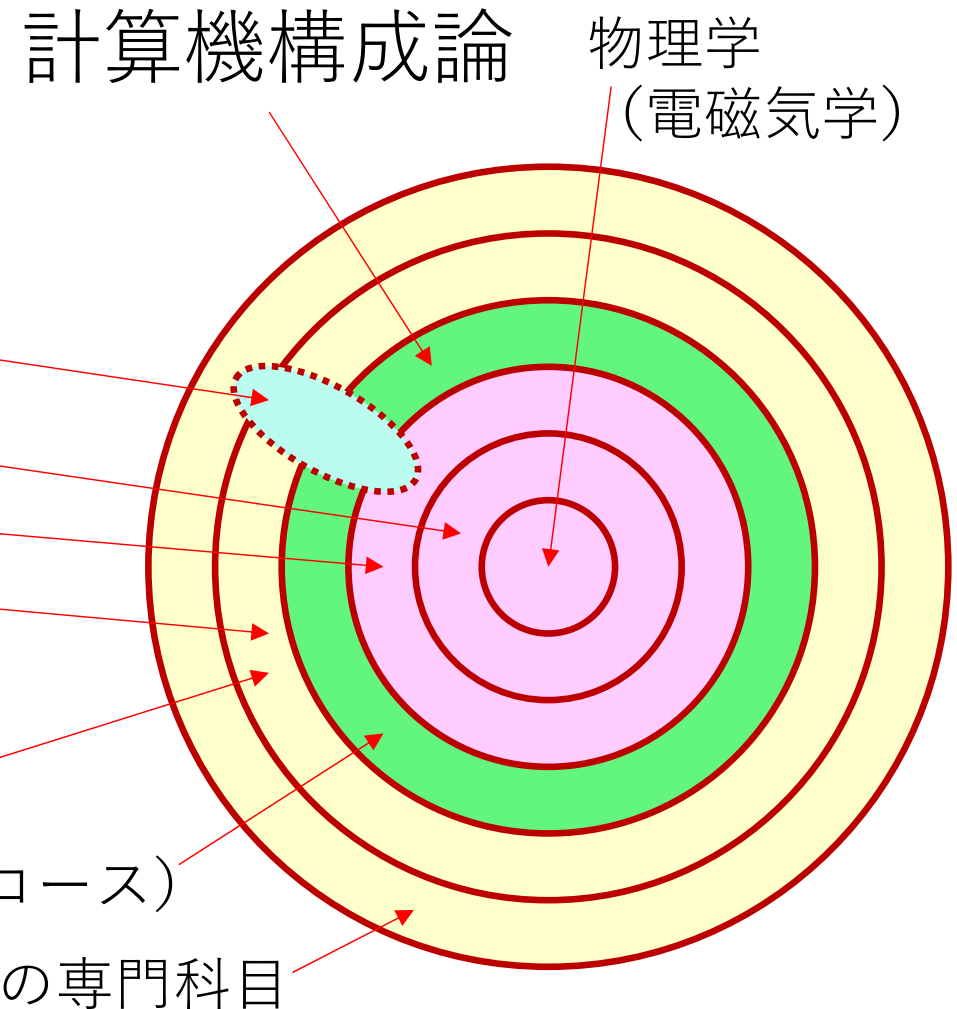
• 先行する関連授業

- 計算機科学入門
- 電気電子回路
- 論理回路
- プログラミング言語

• 後続する関連授業

- オペレーティングシステム
- 計算機アーキテクチャ (SA/SNコース)

情理のその他の専門科目



- ハードウェア科目とソフトウェア科目の間に位置する
- ソフトウェアがハードウェアによってどう実行されるかを学ぶ

「計算機構成論」

(5) 2023年度の授業スケジュール

回	レジュメ	内容
1	Lect 1	イントロダクション: コンピュータシステムの概要
2	Lect 2	コンピュータシステムの構成方法
3	Lect 3	計算機での数の表現, 2進数
4	Lect 4	コンピュータでの計算の実行方法(1) アセンブリ言語、機械語
5		(2) 命令実行制御方式
6	Lect 5	計算機の命令セット (1) MIPS命令セット
7		(2) オペランド指定の方式
8		中間試験 (Lect 1~5) と解説

回	レジュメ	内容
9	Lect 6	手続き・関数呼び出しのハードウェア側での処理
10	Lect 7	C言語から実行可能なプログラムへの変換
11	Lect 8	計算機での算術演算の実現方法 (1) 加算、減算、ALU
12		(2) 乗算
13		(3) 浮動小数点演算
14	Lect 9	コンピュータシステムの性能評価
15		計算機構成論のまとめ
—		期末試験 (Lect 5~9)

期末テストについて

- 試験範囲は Lecture 5～9です
 - これまでの授業の集大成であり、試験範囲全体をおおむねカバーするよう出題されます
 - 中間テスト同様、問題量が多いので、事前の準備(と当日の時間配分?)が重要
- 試験日程、教室、持ち物等はCAMPUS WEBで確認して下さい
 - 遅刻しないこと！
 - 学生証を忘れないこと！
- コースニュースも見ておいて下さい
 - マークシートに解答するため、鉛筆かシャープペンシルを持参して下さい
 - 持ち込み許可物件は無し(水色の紙は無し、電卓もダメ)
 - 問題用紙の1枚目をあらかじめ確認しておくこと

演習 1 :

下図の Func 関数および Sum 関数の Cコードを変換した右図のアセンブリコードの空欄を埋めよ。

```
int Sum(int n) {
    int s = 0;
    int i;
    for (i=-n; i<=n; i++) {
        s += Func(i);
    }
    return s;
}

int Func(int n) {
    if (n>=0) {
        return n;
    }
    else {
        return 0;
    }
}
```

Sum:	addi	\$sp,	\$sp,	-16
	sw	\$ra,	12(\$sp)	
	sw	\$s1,	8(\$sp)	
	sw	\$s2,	4(\$sp)	
	sw	\$s3,	0(\$sp)	
	add	\$s1,	\$a0,	\$zero
	add	\$s2,	\$zero,	\$zero
	sub	\$s3,	\$zero,	\$s1
Loop:	slt	\$t0,	\$s1,	\$s3
	bne	\$t0,	\$zero,	Exit
	add	\$a0,	\$s3,	\$zero
	(a)	Func		
	add	\$s2,	\$s2,	(b)
	addi	\$s3,	\$s3,	1
	j	Loop		
Exit:	add	\$v0,	\$s2,	\$zero
	lw	(c)	0(\$sp)	
	lw	(d)	4(\$sp)	
	lw	(e)	8(\$sp)	
	lw	(f)	12(\$sp)	
	addi	\$sp,	\$sp,	(g)
	(h)	(i)		

Func:	slt	\$t0,	\$a0,	\$zero
	(j)	\$t0,	\$zero,	Else
	add	\$v0,	\$zero,	\$a0
	jr	\$ra		
Else:	add	\$v0,	\$zero,	\$zero
	jr	\$ra		

演習 2 :

オブジェクトファイル main.o および proc1.o がそれぞれ右図のように与えられているとき、これらをリンクした下図の実行ファイルの空欄にあてはまる数値を16進数で答えよ。但し、下図のアドレス(c)の jal 命令は main.o の jal 命令に対応し、下図のアドレス(e)および(g)の lw 命令および sw 命令はそれぞれ、proc1.o の lw 命令および sw 命令に対応するものとする。また、\$gp の値は 10008000₁₆ とせよ。

リンク結果		
ヘッダ	テキスト・サイズ	(a)
	データ・サイズ	(b)
テキスト・セグメント	アドレス	
	00400000 ₁₆	lw \$t0, 8000 ₁₆ (\$gp)

	(c)	jal (d)

	(e)	lw \$t0, (f)(\$gp)

	(g)	sw \$t1, (h)(\$gp)
データ・セグメント
	10000000 ₁₆	(X)

	(i)	(Y)
	(j)	(Z)

main.o		名前	main	
ヘッダ	テキスト・サイズ	データ・サイズ	500 ₁₆	
		データ・サイズ	50 ₁₆	
テキスト・セグメント	アドレス			
	0 ₁₆	lw \$t0, 0(\$gp)		
		
	30 ₁₆	jal 0		
データ・セグメント	0 ₁₆	(X)		
		
リロケーション情報	アドレス	命令タイプ	依存関係	
	0 ₁₆	lw	X	
	30 ₁₆	jal	proc1	
シンボル表	ラベル	アドレス		
	X	-		
	proc1	-		

proc1.o		名前	proc1	
ヘッダ	テキスト・サイズ	データ・サイズ	300 ₁₆	
		データ・サイズ	30 ₁₆	
テキスト・セグメント	アドレス			
	0 ₁₆	lw \$t0, 0(\$gp)		
		
	70 ₁₆	sw \$t1, 0(\$gp)		
データ・セグメント	0 ₁₆	(Y)		
	4 ₁₆	(Z)		
		
リロケーション情報	アドレス	命令タイプ	依存関係	
	0 ₁₆	lw	Y	
	70 ₁₆	sw	Z	
シンボル表	ラベル	アドレス		
	Y	-		
	Z	-		
	...	-		

演習 3 :

Lecture 8 で扱った「最終バージョンの乗算アルゴリズム」に沿って、4ビットの被乗数 7_{10} と乗数 5_{10} の積を求める過程を書け。

サイクル	ステップ	被乗数レジスタ	積レジスタ
0	初期値	0111	
1	積レジスタのLSBが1なら被乗数を積レジスタの左半分に足す	0111	
	積レジスタを右シフト	0111	
2	積レジスタのLSBが1なら被乗数を積レジスタの左半分に足す	0111	
	積レジスタを右シフト	0111	
3	積レジスタのLSBが1なら被乗数を積レジスタの左半分に足す	0111	
	積レジスタを右シフト	0111	
4	積レジスタのLSBが1なら被乗数を積レジスタの左半分に足す	0111	
	積レジスタを右シフト	0111	

演習 4 :

- (1) 10進数の 1.0 および 2.0 を IEEE-754規格の単精度浮動小数点形式で表現した時の32ビットのビット列をそれぞれ8桁の16進数で表せ。
- (2) 10進数の 1.0 および 2.0 を IEEE-754規格の倍精度浮動小数点形式で表現した時の64ビットのビット列をそれぞれ16桁の16進数で表せ。
- (3) 10進数の 256.5 を IEEE-754規格の単精度浮動小数点形式で表現した時の32ビットのビット列を8桁の16進数で表せ。
- (4) 10進数の -65536.125 を IEEE-754規格の倍精度浮動小数点形式で表現した時の64ビットのビット列を16桁の16進数で表せ。