第 4 回 実験レポート

森田 蓮

2025 年 1 月 9 日

1 目的

本実験の目的は以下の通りである．

1. コンパイラを使わずに，アセンブリ言語・機械語によって整列アルゴリズムを直接記述することが可能であることを示す．
2. 疑似コードや高級言語での実装と比べて，大きく異なる点があるかどうかを明らかにする．また，コードの複雑さやコード量がどの程度増すかを明らかにする．
3. 高級言語での実装と比べて計算時間に差があるかどうかを明らかにする．

2 方法

初めにアセンブリ言語で整列アルゴリズムを実装する．次にアセンブリ言語で実装した整列アルゴリズムが適切な動作をするか入力データを与え実行されるか確認する．

また，アセンブリ言語で実装した整列アルゴリズムの疑似コードを高級言語 (java) で実装しコードの複雑さやコード量の変化の比較対象とする．以下の 4 つのデータセットを使用し，コードの実行時に time のコマンドを付けて実行し互いの時間差を計測する．

- ソート済みデータ: $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$
- 逆順データ: $\{10, 9, 8, 7, 6, 5, 4, 3, 2, 1\}$
- ランダムデータ: $\{45, 12, 78, 34, 56, 255, 100, 23\}$
- 重複データ: $\{5, 1, 5, 2, 5, 3, 5, 4, 5\}$

3 結果

実験の結果，アセンブリ言語で記述した整列アルゴリズムは正しく動作することが確認された．

高級言語 (java) で記述したバブルソートと比較した場合，アセンブリ言語で実装したコードは

ソートするするまでの工数が多く複雑であることがわかる．

5 回実行し平均の計算時間の差を以下の表にまとめる．

データセット	Java の実行時間 (ms)	アセンブリの実行時間 (ms)
ソート済みデータ	0.021s	0.001s
逆順データ	0.023s	0.001s
ランダムデータ	0.023s	0.001s
重複データ	0.024s	0.001s

表 1　バブルソートの実行時間比較

4　考察

本実験を通して，アセンブリ言語で整列アルゴリズムを実装することは可能であることが確認された．しかし，java のような高級言語に比べると記述に縛りがなく自由度が高い．それ故にコードが複雑でコード量が増加する点が課題である．結果よりアセンブリ言語の実行時間が全てのデータセットで Java よりも短いことがわかった．また，アセンブリ言語での実装はハードウェアに依存するため，特定の環境で最適化を行った場合，さらに計算時間が短縮される可能性がある．

参考文献

GeeksforGeeks, Bubble Sort Algorithm, https://www.geeksforgeeks.org/bubble-sort-algorithm/, Accessed: 2024-12-27

付録

アセンブリ言語:

```
    section .data
    data:  dd 45, 12, 78, 34, 56, 255,
         100, 23   ; 配列データ
    ndata equ 8  ; 配列の要素数
section .text
    global _start

_start:
    mov ecx, data
    mov esi, ndata

Outloop:
    dec esi
    jz end              ;ecx = 0 で終了
    mov edi, ecx
    mov edx, esi

Inloop:
    mov eax, [edi]          ;今の値
    mov ebx, [edi + 4]      ;次

    cmp eax, ebx
    jbe Noswap        ; eax >= ebx
        Noswap

```

```
    mov [edi], ebx
    mov [edi + 4], eax

Noswap:
    add edi, 4
    dec edx
    jnz Inloop
    jmp Outloop

end:
    mov eax, 1
    mov ebx, 0
    int 0x80
```

Java コード:

```java
public class BubbleSort {
    public static void main(String[]
        args) {
        int[] data = {45, 12, 78, 34,
            56, 255, 100, 23};

        bubbleSort(data);

        System.out.println("0");
    }

public static void bubbleSort(int[]
    array) {
    int n = array.length;
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - 1 - i;
             j++) {
            if (array[j] > array[j +
                1]) {
                // Swap array[j] and
                    array[j+1]
                int temp = array[j];
                array[j] = array[j +
                    1];
                array[j + 1] = temp;
                }
            }
        }
    }
}
```