**ĐẠI HỌC QUỐC GIA TP HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

Logo, company name

Description automatically generated

**KIẾN TRÚC MÁY TÍNH(CO2008)**

**BÁO CÁO BÀI TẬP LỚN**

ĐỀ TÀI: SELECTION SORT CHO SỐ NGUYÊN

**Giảng viên hướng dẫn: Th.s Nguyễn Xuân Minh**

|  |  |
| --- | --- |
| **Sinh viên thực hiện** | **Mã số sinh viên** |
| Phạm Duy Anh | 2310139 |
| Nguyễn Hoàng Gia Bảo | 2310242 |

**Mục lục**

[I.Lý thuyết. 2](#_Toc182573192)

[II.Giải pháp hiện thực. 2](#_Toc182573193)

[III. Giải thuật. 2](#_Toc182573194)

[IV.Thống kế số lệnh, loại lệnh đã sử dụng. 3](#_Toc182573195)

[1. Nhóm lệnh số học và luận lý 3](#_Toc182573196)

[2. Nhóm lệnh di chuyển dữ liệu 3](#_Toc182573197)

[3. Nhóm lệnh hỗ trợ ra quyết định 4](#_Toc182573198)

[4. Các lệnh khác 4](#_Toc182573199)

[5. Tóm tắt số lượng lệnh theo nhóm 4](#_Toc182573200)

[IV. Thời gian chạy của chương trình. 4](#_Toc182573201)

[V. Kết quả kiểm thử. 5](#_Toc182573202)

# I.Lý thuyết.

Thuật toán Selection Sort sắp xếp một mảng bằng cách liên tục tìm phần tử nhỏ nhất (xét theo thứ tự tăng dần) từ phần chưa được sắp xếp và đặt nó ở đầu. Thuật toán duy trì hai mảng con trong một mảng nhất định.

1. Mảng con đã được sắp xếp.
2. Mảng con còn lại chưa được sắp xếp.

Trong mỗi lần lặp lại sắp xếp lựa chọn, phần tử tối thiểu (xét theo thứ tự tăng dần) từ mảng con chưa được sắp xếp sẽ được chọn và chuyển đến mảng con đã sắp xếp.

# II.Giải pháp hiện thực.

Chương trình mở một file nhị phân có tên là INT10.BIN và đọc 10 phần tử nguyên từ file đó vào mảng arr trong bộ nhớ. Sau đó, chương trình in mảng trước khi sắp xếp, sắp xếp mảng bằng thuật toán Selection Sort, và in ra mảng đã được sắp xếp.

Các phần chính của chương trình bao gồm:

* Mở file và đọc dữ liệu vào mảng arr.
* In mảng trước khi sắp xếp và sau khi sắp xếp.
* Hàm Selection Sort sắp xếp mảng từ nhỏ đến lớn.
* Hàm mini tìm chỉ số phần tử nhỏ nhất trong mảng từ vị trí hiện tại đến cuối mảng.
* Hàm swap hoán đổi hai phần tử trong mảng.

# III. Giải thuật.

Giải thuật Selection Sort hoạt động bằng cách tìm phần tử nhỏ nhất từ một vị trí trong mảng đến cuối mảng và hoán đổi nó với phần tử ở vị trí đầu tiên chưa được sắp xếp. Quá trình này được lặp lại cho đến khi toàn bộ mảng được sắp xếp.

Chi tiết từng bước:

1. Duyệt mảng từ vị trí i = 0 đến vị trí size - 2.
2. Tại mỗi bước i, tìm phần tử nhỏ nhất trong đoạn mảng từ i đến size - 1 (gọi hàm mini để tìm chỉ số của phần tử nhỏ nhất).
3. Hoán đổi phần tử nhỏ nhất vừa tìm được với phần tử tại vị trí i (gọi hàm swap để hoán đổi).
4. Sau mỗi bước sắp xếp, chương trình in mảng để theo dõi các bước sắp xếp.

Giải thích chi tiết các hàm

* Hàm **selectionSort**: thực hiện thuật toán Selection Sort bằng cách gọi lần lượt các hàm mini và swap để tìm và hoán đổi phần tử nhỏ nhất với phần tử đầu tiên chưa được sắp xếp.
* Hàm **mini**: tìm chỉ số của phần tử nhỏ nhất từ một vị trí trong mảng đến cuối mảng. Hàm trả về chỉ số của phần tử này qua thanh ghi $v0.
* Hàm **swap**: thực hiện hoán đổi hai phần tử trong mảng dựa trên chỉ số được truyền vào thông qua thanh ghi $a1 (chỉ số đầu) và $a2 (chỉ số cuối).

# IV.Thống kế số lệnh, loại lệnh đã sử dụng.

## 1. Nhóm lệnh số học và luận lý

| **Lệnh** | **Cú pháp sử dụng trong bài** | **Ý nghĩa** | **Số lần** |
| --- | --- | --- | --- |
| add | add $t3, $t0, $t2 | Cộng hai thanh ghi và lưu vào thanh ghi đích. | 3 |
| addi | addi $v0, $zero, 4 | Cộng một giá trị hằng vào thanh ghi. | 10 |
| sll | sll $t2, $s4, 2 | Dịch trái (nhân với lũy thừa của 2). | 6 |
| subi | subi $s2, $a1, 1 | Trừ giá trị hằng từ thanh ghi. | 1 |
| bge | bge $s1, $s2, iForExit | Nhảy nếu lớn hơn hoặc bằng. | 2 |
| bgt | bgt $t5, $t2, miniEnd | Nhảy nếu lớn hơn. | 1 |

**Tổng cộng:** 23 lệnh.

## 2. Nhóm lệnh di chuyển dữ liệu

| **Lệnh** | **Cú pháp sử dụng trong bài** | **Ý nghĩa** | **Số lần** |
| --- | --- | --- | --- |
| sw | sw $ra, 0($sp) | Lưu dữ liệu từ thanh ghi vào bộ nhớ. | 25 |
| lw | lw $a0, 0($t3) | Tải dữ liệu từ bộ nhớ vào thanh ghi. | 6 |
| li | li $v0, 1 | Tải giá trị hằng vào thanh ghi. | 7 |
| move | move $t0, $v0 | Sao chép giá trị từ thanh ghi này sang thanh ghi khác. | 8 |

**Tổng cộng:** 26 lệnh.

## 3. Nhóm lệnh hỗ trợ ra quyết định

| **Lệnh** | **Cú pháp sử dụng trong bài** | **Ý nghĩa** | **Số lần** |
| --- | --- | --- | --- |
| beqz | beqz $t1, closeFile | Nhảy nếu thanh ghi bằng 0. | 2 |
| bltz | bltz $t0, errorOpenFile | Nhảy nếu thanh ghi nhỏ hơn 0. | 2 |
| bgez | bgez $t0, successOpenFile | Nhảy nếu thanh ghi lớn hơn hoặc bằng 0. | 1 |

**Tổng cộng:** 5 lệnh.

## 4. Các lệnh khác

| **Lệnh** | **Cú pháp sử dụng trong bài** | **Ý nghĩa** | **Số lần** |
| --- | --- | --- | --- |
| j | j readFile | Nhảy không điều kiện đến nhãn. | 5 |
| jal | jal selectionSort | Gọi hàm con (lưu địa chỉ trở về). | 4 |
| jr | jr $ra | Trở về địa chỉ lưu trong thanh ghi $ra. | 4 |
| syscall | syscall | Gọi hệ thống (I/O, xử lý file, v.v.). | 15 |

**Tổng cộng:** 28 lệnh.

## 5. Tóm tắt số lượng lệnh theo nhóm

| **Nhóm lệnh** | **Số lệnh** |
| --- | --- |
| Nhóm số học và luận lý | 23 |
| Nhóm di chuyển dữ liệu | 26 |
| Nhóm hỗ trợ ra quyết định | 5 |
| Các lệnh khác | 28 |

**Tổng số lệnh:** **82** lệnh.

# IV. Thời gian chạy của chương trình.

Với:

* Các lệnh như add, addi, subi, move, và sll mất 1 chu kỳ mỗi lệnh.
* Các lệnh lw, sw, và li mất 2 chu kỳ mỗi lệnh.
* Các lệnh j, jal, jr, và các lệnh nhảy có điều kiện (beqz, bltz, bgez, ...) mất 2 chu kỳ mỗi lệnh.
* Lệnh syscall mất 8 chu kỳ do phải thực hiện các thao tác vào/ra.

Dựa vào thống kê các lệnh trong chương trình, ta tính tổng số chu kỳ:

T = 1\*23 + 2\*52 + 2\*10 + 2\*13 + 8\*15 = 219 (chu kỳ)

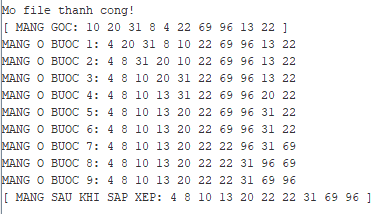
Thời gian chạy của chương trình:

CPU Time = = = 2.19 \* (s)

# V. Kết quả kiểm thử.

Với file INT10.BIN là mảng gồm 10 phần tử [10,20,31,8,4,22,69,96,13,22]

Sau khi biên dịch chương trình, thu lại được kết quả sau:



**Nhận xét:** Đúng.