**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KỸ THUẬT MÁY TÍNH**

**TỔ CHỨC VÀ CẤU TRÚC MÁY TÍNH II (IT012)**

**A blue logo with a planet and text

Description automatically generated**

**Sinh viên: Phan Lê Minh**

**MSSV: 23520952**

**Giảng viên hướng dẫn: Nguyễn Thành Nhân**

**IT012.O21.1 – Lab 03**

**Mục lục**

[1. Lý thuyết 4](#_Toc163418053)

[2. Thực hành 4](#_Toc163418054)

[2.1. Mô phỏng thực thi các lệnh cơ bản 4](#_Toc163418055)

[2.2. Mô phỏng và nêu ý nghĩa các chương trình 5](#_Toc163418056)

[3. Bài tập 9](#_Toc163418057)

[3.1. Nhập vào một chuỗi, xuất ra cửa sổ I/O của MARS theo từng yêu cầu 9](#_Toc163418058)

[**a)** **Khai báo và xuất ra cửa sổ I/O 2 chuỗi có giá trị cho trước** 9](#_Toc163418059)

[**b)** **Biểu diễn nhị phân của 2 chuỗi trên dưới bộ nhớ** 10](#_Toc163418060)

[**c)** **Xuất ra lại đúng chuỗi đã nhập** 10](#_Toc163418061)

[**d)** **Nhập vào 2 số nguyên, sau đó xuất ra tổng 2 số nguyên này** 12](#_Toc163418062)

**Đề mục Viết tắt – Kí hiệu – Thuật ngữ**

|  |  |  |
| --- | --- | --- |
| **Kí hiệu / thuật ngữ** | **Ý nghĩa** | **Ghi chú** |
| rd, rs, rt | Các thành phần trên thanh ghi |  |
| \* | Con trỏ *(pointer)* | Mặc định những cái tên “rd”, “rs”, “rt” khi không có con trỏ là tên biến |
| Imm | Số tức thời *(số)* | Immediate |
| SignImm | Số có dấu tức thời | Sign Immediate |
| (1) | Ghi chú (1) số SignImm và Imm được dịch sang hệ bù 2 rồi mới tiến hành tính toán |  |
| Standard input | Bộ nhập chuẩn *(bàn phím)* |  |
| Standard ouput | Bộ xuất chuẩn *(màn hình)* |  |
| Syscall | Lệnh gọi hệ thống thực thi tập lệnh định sẵn |  |

**Kiểu dữ liệu**

|  |  |  |
| --- | --- | --- |
| **Kiểu dữ liệu** | **Giải thích** | **Ghi chú** |
| word | Chuỗi dữ liệu kí tự | 1 word = 4 bytes |
| space | Khoảng trống *(ô)* | 1 space = 1 byte |

1. **Lý thuyết**

* Giảng viên hướng dẫn sinh viên sử dụng phần mềm MARS dựa theo tài liệu: **MARS – chương trình mô phỏng hợp ngữ (assembly) MIPS.**

1. **Thực hành**
   1. **Mô phỏng thực thi các lệnh cơ bản**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên lệnh** | **Cú pháp** | **Diễn giải tóm tắt** | **Diễn giải chi tiết** |
| add | add rd, rs, rt | rd = rs + rt | Lấy tổng giá trị của rs, rt đem gán vào rd, có báo tràn bit |
| addi | addi rt, rs, Imm | rt = rs + SignImm | Lấy tổng giá trị của rs, SignImm đem gán vào rd, có báo tràn bit |
| addu | addu rd, rs, rt | rd = rs + rt | Lấy tổng giá trị của rs, rt đem gán vào rd, không báo tràn bit |
| addiu | addiu rd, rs, Imm | rt = rs + Imm | Lấy tổng giá trị của rs, Imm đem gán vào rd, không báo tràn bit |
| sub | sub rd, rs, r | rd = rs - rt | Lấy hiệu của rs và rt gán vào rd, có báo tràn bit |
| subu | subu rd, rs, rt | rd = rs - rt | Lấy hiệu của rs và rt gán vào rd, không báo tràn bit |
| and | and rd, rs, rt | rd = (rs&rt) | Thực hiện toán tử AND trên các bit của rs, rt và gán vào rd |
| andi | andi rt, rs, Imm | rt = (rs&Imm) | Thực hiện toán tử AND trên các bit của rs, Imm và gán vào rt |
| or | or rd, rs, rt | rd = (rs|rt) | Thực hiện toán tử OR trên các bit của rs, rt và gán vào rd |
| nor | nor rd, rs, rt | rd = ~(rs|rt) | Thực hiện toán tử NOR trên các bit của rs, rt và gán vào rd |
| lw | lw rt, Imm(rs) | \*rt = \*(rs + Imm) | Lấy giá trị của ô nhớ thứ Imm tính từ rs lưu vào rt |
| sw | sw rt, Imm(rs) | \*(rs + Imm) = \*Rt | Lấy giá trị của rt lưu vào ô nhớ thứ Imm tính từ rs |
| slt | slt rd, rs, rt | rd = (rs < rt)? 1 : 0 | rd bằng 0 nếu rs < rt, bằng 1 nếu rs >= rt, thực hiện trên số có dấu |
| slti (1) | slti rt, rs, SignImm | rt = (rs < SignImm)? 1 : 0 | rd bằng 0 nếu rs < SignImm, bằng 1 nếu rs >= SignImm |
| sltu | sltu rd, rs, rt | rd = (rs < rt)? 1 : 0 | rd bằng 0 nếu rs < rt, bằng 1 nếu rs >= rt, thực hiện trên số không dấu |
| sltiu (1) | sltiu rt, rs, Imm | rt = (rs < Imm)? 1 : 0 | rd bằng 0 nếu rs < Imm, bằng 1 nếu rs >= Imm |
| syscall | syscall |  | Thực hiện lệnh định sẵn của hệ điều hành |

* 1. **Mô phỏng và nêu ý nghĩa các chương trình**
     1. **Ví dụ 1**

**A screenshot of a computer program

Description automatically generated**

****

|  |  |
| --- | --- |
| **Line** | **Giải thích** |
| **2** | Tạo 1 biến số nguyên tên **var1**, chiếm 1 **word** lưu trữ, với giá trị khởi tạo bằng **23** |
| **5** | Lệnh **lw** thực hiện việc load giá trị **var1** vào thanh ghi **$t0** |
| **6** | Lệnh **li** đưa số **5** vào thanh ghi **$t1** |
| **7** | Lệnh **sw** lưu giá trị của thanh ghi **$t1** vào **var1** |

* + 1. **Ví dụ 2**

**A screenshot of a computer program

Description automatically generated**

****

|  |  |
| --- | --- |
| **Line** | **Giải thích** |
| **2** | Tạo 1 mảng tên **array1**, có **12 bytes (3 words)** liên tục với nhau lưu trữ liên tục chưa khởi tạo. |
| **6** | Lệnh **la** load địa chỉ **array1** vào thanh ghi **$t0** |
| **7** | Lệnh **li** đưa số **5** vào thanh ghi **$t1** |
| **8** | Lệnh **sw** lưu giá trị của thanh ghi **$t1** vào từ nhớ có địa chỉ **$t0** (địa chỉ của phần tử đầu tiên trong mảng **array1)** |
| **9** | Lệnh **li** đưa số **13** vào thanh ghi **$t1** |
| **10** | Lệnh **sw** lưu giá trị của thanh ghi **$t1** vào từ nhớ có địa chỉ **$t0+4** (địa chỉ của phần tử kế tiếp trong mảng **array1)** |
| **11** | Lệnh **li** đưa số **-7** vào thanh ghi **$t1** |
| **12** | Lệnh **sw** lưu giá trị của thanh ghi **$t1** vào từ nhớ có địa chỉ **$t0+8** (địa chỉ của phần tử cuối cùng trong mảng **array1)**   * **array1 = [5, 13, -7]** |

* + 1. **Ví dụ 3**

**A screenshot of a computer screen

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

|  |  |
| --- | --- |
| **Line** | **Giải thích** |
| **1** | Lệnh **li** đưa số **5** vào thanh ghi **$v0** |
| **2** | Lệnh **syscall** gọi hàm hệ thống tương ứng với giá trị số (=5) đang lưu trữ trong **$v0 (đọc số nguyên)** |

* + 1. **Ví dụ 4**

**A screenshot of a computer program

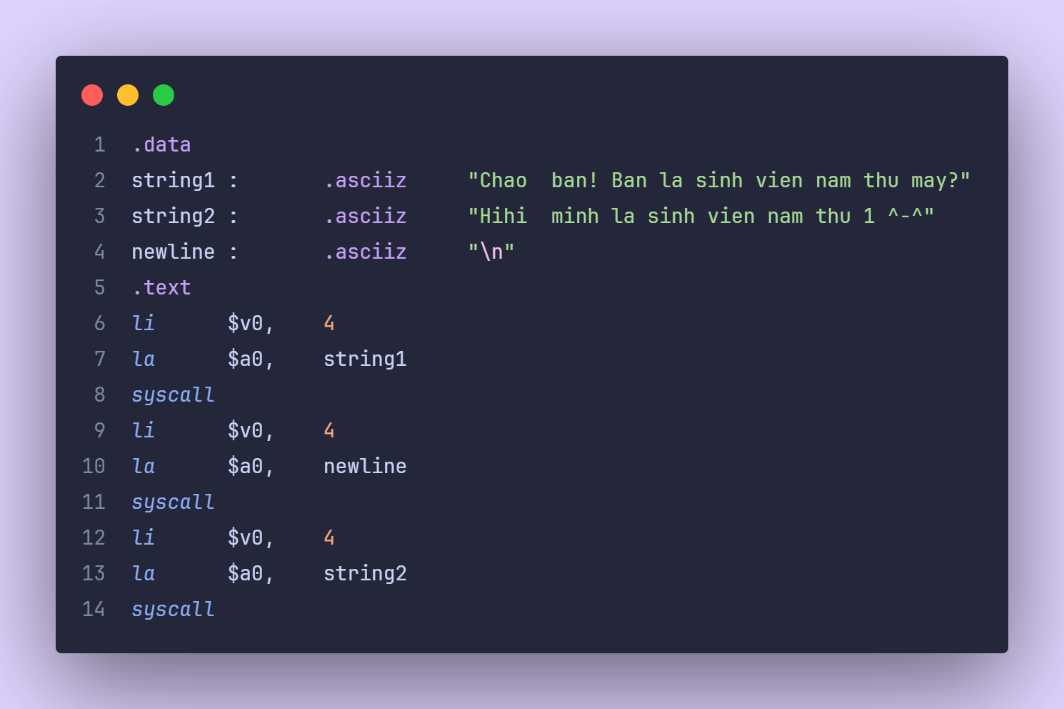
Description automatically generated**

**A screenshot of a computer

Description automatically generated**

|  |  |
| --- | --- |
| **Line** | **Giải thích** |
| **2** | Tạo 1 chuỗi tên **string1** với khởi tạo “Print this. \n” |
| **5** | Điểm bắt đầu chương trình |
| **6** | Lệnh **li** đưa số **4** vào thanh ghi **$v0** |
| **7** | Lệnh **la** load địa chỉ **string1** vào thanh ghi **$a0** |
| **8** | Lệnh **syscall** gọi hàm hệ thống tương ứng với giá trị số (=4) đang lưu trữ trong **$v0 (in chuỗi có địa chỉ được lưu trong $a0)**   * **In ra “Print this.” và xuống dòng** |

1. **Bài tập**
   1. **Nhập vào một chuỗi, xuất ra cửa sổ I/O của MARS theo từng yêu cầu**
2. **Khai báo và xuất ra cửa sổ I/O 2 chuỗi có giá trị cho trước**

****

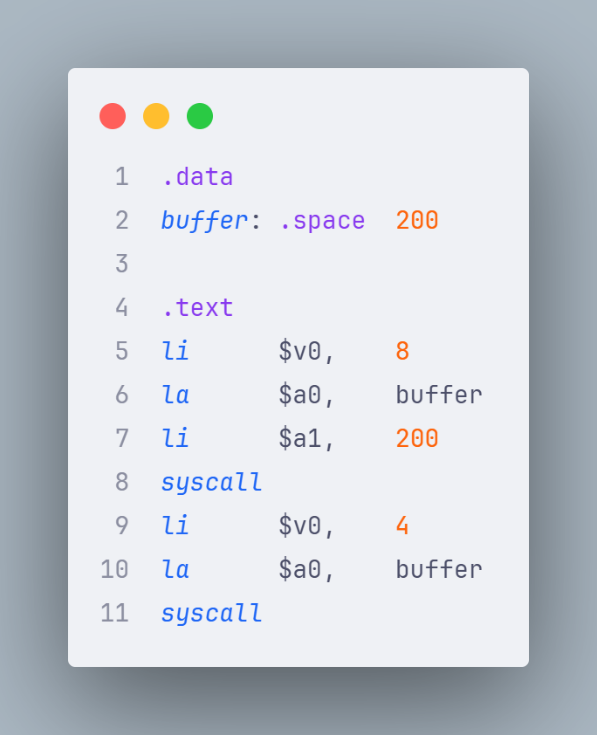
****

|  |  |
| --- | --- |
| **Line** | **Giải thích** |
| **2** | **string1** là biến trong vùng nhớ có kiểu dữ liệu **asciiz**, có giá trị “Chao ban! Ban la sinh vien nam thu may?” |
| **3** | **string2** là biến trong vùng nhớ có kiểu dữ liệu **asciiz**, có giá trị “Hihi minh la sinh vien nam thu 1 ^-^” |
| **4** | **newline** là biến trong vùng nhớ có kiểu dữ liệu **asciiz**, có giá trị “\n” *(kí tự xuống dòng)* |
| **6** | Gán giá trị 4 vào **$v0** |
| **7** | Gán địa chỉ của **string1** vào **$a0** |
| **8** | Syscall gọi xuất ra standard output giá trị của **$a0** *(tức* ***string1****)* |
| **9** | Gán giá trị 4 vào **$v0** |
| **10** | Gán địa chỉ của **newline** vào **$a0** |
| **11** | Syscall gọi xuất ra standard output giá trị của **$a0** *(tức* ***newline****)* |
| **12** | Gán giá trị 4 vào **$v0** |
| **13** | Gán địa chỉ của **string2** vào **$a0** |
| **14** | Syscall gọi xuất ra standard output giá trị của **$a0** *(tức* ***string2****)* |

1. **Biểu diễn nhị phân của 2 chuỗi trên dưới bộ nhớ**
2. **Xuất ra lại đúng chuỗi đã nhập**

**A screenshot of a computer program

Description automatically generated**

****

|  |  |
| --- | --- |
| **Line** | **Giải thích** |
| **2** | Tạo khoảng trống trong bộ nhớ có kích thước 100 Bytes cho **buffer** |
| **5** | Gán giá trị 8 cho **$v0** |
| **6** | Gán địa chỉ của **buffer** vào **$a0** |
| **7** | Gán giá trị 200 vào **$a1** |
| **8** | Syscall gọi nhập chuỗi kí tự từ standard input *(bàn phím)* với kích thước tối đa là 100. Sau đó lưu vào **$a0** *(tức* ***buffer****)* |
| **9** | Gán giá trị 4 vào **$v0** |
| **10** | Gán địa chỉ của **buffer** vào **$a0** |
| **11** | Syscall gọi in giá trị chuỗi từ **$a0** *(tức* ***buffer****)* |

1. **Nhập vào 2 số nguyên, sau đó xuất ra tổng 2 số nguyên này**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

|  |  |
| --- | --- |
| **Line** | **Giải thích** |
| **1** | Gán giá trị 5 vào **$v0** |
| **2** | Syscall gọi nhập số từ standard input. Sau đó sẽ gán ngược lại vào **$v0** |
| **3** | Gán giá trị **$zero** *(0)* + **$v0** *(input lần 1)* vào **$a0** |
| **5** | Gán giá trị 5 vào **$v0** |
| **6** | Syscall gọi nhập số từ standard input *(bàn phím)*. Sau đó sẽ gán ngược lại vào **$v0** |
| **7** | Gán giá trị **$a0** *(line 3)* + **$v0** *(input lần 2)* vào **$a0** |
| **9** | Gán giá trị 1 vào **$v0** |
| **10** | Syscall gọi in giá trị của **$a0** ra standard output |