

ASSIGNMENT #3

Computer Hardware

ECE 314/13CTT

Nguyễn Tuấn Nam - Lương Việt Thắng - Nguyễn Văn Linh

Ngày 25 tháng 3 năm 2015

1 Mục tiêu

- Lập trình cơ bản trong MIPS
- Hiểu các quy tắc gọi hàm, sử dụng mảng, chuỗi, stack ... trong MIPS.
- Thao tác với file trong MIPS.
- Cài đặt các phép toán trên ma trận 2 chiều.

2 Nội dung

Cài đặt các hàm thao tác căn bản trên ma trận dưới dạng ngôn ngữ MIPS

.

```
1 int checkMatrixIdentity ( int * arr1, int m_arr, int n_arr) ;
```

- Mô tả:
 - Kiểm tra ma trận có phải là ma trận đơn vị.
- Tham số:
 - `arr`: con trỏ trỏ tới ma trận với kích cỡ $m \times n$.
 - `m_arr`: số dòng của `arr`.
 - `n_arr`: số cột của `arr`.
- Giá trị trả về:
 - Trả về 1 nếu ma trận đầu vào có dạng giá trị đường chéo tất cả là 1 và phần còn lại của ma trận có giá trị là 0.
 - Ngược lại trả về 0.

```
1 int* transMatrix ( int * arr1, int m_arr, int n_arr) ;
```

- Mô tả:
 - Chuyển vị ma trận (transpose).
- Tham số:
 - arr: con trỏ trỏ tới ma trận với kích cỡ $m \times n$.
 - m_arr: số dòng của arr.
 - n_arr: số cột của arr.
- Giá trị trả về:
 - Trả về ma trận có kích thước $n \times m$ là chuyển vị của ma trận đầu vào.

```
1 int* plusMatrix ( int * arr1, int m_arr1, int n_arr1, int * arr2, int
    m_arr2, int n_arr2) ;
```

- Mô tả:
 - Cộng hai ma trận.
- Tham số:
 - arr1: con trỏ trỏ tới ma trận thứ 1 với kích cỡ $m \times n$.
 - m_arr1: số dòng của arr1.
 - n_arr1: số cột của arr1.
 - Các tham số arr2, m_arr2, n_arr2 có ý nghĩa tương tự.
- Giá trị trả về:
 - Nếu hai ma trận arr1 và arr2 có thể cộng được thì trả về một ma trận là tổng của 2 ma trận đầu vào, kích cỡ như cũ.
 - Ngược lại trả về con trỏ NULL.

```
1 int* multiMatrixConst ( int * arr, int m_arr, int n_arr, int val) ;
```

- Mô tả:
 - Nhân ma trận với hằng số.
- Tham số:

- `arr`: con trỏ trỏ tới ma trận thứ 1 với kích cỡ $m \times n$.
- `m_arr`: số dòng của `arr`.
- `n_arr`: số cột của `arr`.
- `val`: hằng số để nhân với ma trận.
- Giá trị trả về:
 - Trả về ma trận có kích cỡ $m \times n$ với mỗi giá trị ở vị trí (i, j) được tính bằng $arr[i, j] \times val$.

Cấu trúc file input: Cấu trúc file **nhị phân** dùng để chạy chương trình như sau:

- 2 số nguyên đầu tiên:
 1. Số nguyên đầu tiên m_A chỉ ra số dòng của ma trận A .
 2. Số nguyên thứ hai n_A chỉ ra số cột của ma trận A .
- $m_A * n_A$ số nguyên tiếp theo là các phần tử của ma trận A được liệt kê từ trái qua phải và từ trên xuống dưới.
- 2 số nguyên tiếp theo:
 1. Số nguyên đầu tiên m_B chỉ ra số dòng của ma trận B .
 2. Số nguyên thứ hai n_B chỉ ra số cột của ma trận B .
- $m_B \times n_B$ số nguyên tiếp theo là các phần tử của ma trận B được liệt kê từ trái qua phải và từ trên xuống dưới.

Lưu ý: File input mẫu đính kèm.

Cài đặt chương trình minh họa các hàm đã cài đặt. Chương trình minh họa có giao diện như sau:

- Nhập đường dẫn đến file input.
- Nhập hằng số.
- Xuất ra các kết quả sau:
 - Kiểm tra có là ma trận đơn vị.
 - Ma trận chuyển vị.
 - Tổng 2 ma trận.
 - Tích ma trận với hằng số.

Lưu ý: Không được tạo menu chọn, chỉ cần xuất ra 4 kết quả theo thứ tự đã nêu trên.

3 Yêu cầu

- Sử dụng chương trình MARS.
- Các hàm trên không được sử dụng biến toàn cục để lưu trữ giá trị tính toán trung gian, phải sử dụng biến cục bộ.
- Sinh viên nộp bài theo cấu trúc sau:
 - Thư mục Source: chứa mã nguồn của chương trình (file .asm).
 - Thư mục Doc: chứa file mô tả cách thức cài đặt các hàm quan trọng.