Bài thực hành số 6

Lớp: 139365 – Học phần: Thực Hành Kiến Trúc Máy Tính

Họ và Tên: Nguyễn Anh Thứ MSSV: 20215144

Bài 1:

Text

Description automatically generated

Text, letter

Description automatically generated

.data: lưu dữ liệu đề bài

Main: gán địa chỉ đầu của mảng A cho thanh ghi a0



Gán số phần tử của A cho a1



Mspfx: định nghĩa cho v0 là số phần tử của mảng con lớn nhất, v1 là tổng của mảng con lớn nhất, t0 và t1 lần lượt là số phần tử i và tổng mảng con của biến đếm. Cả 4 giá trị được đặt là 0 với giá trị ban đầu.

Loop: với 3 lệnh add đầu, ta có được giá trị của địa chỉ của phần tử A[i] bằng cách lấy tổng của a0 + 4i và lưu vào thanh ghi t3 (bởi các phần tử của mảng ở đây là số nguyên nên chiếm 4 byte lưu trữ.)



Lw: đọc vào t4 giá trị của A[i] qua địa chỉ của nó ở t3



Add tiếp: cộng t4 vào t1 để tạo ra tổng tiếp theo



Slt: so sánh v1 và t1, với v1 là tổng max cũ và t1 là tổng tiếp. nếu v1 < t1, ta được giá trị 1 ở t5 và ngược lại

Bne: nếu t5 khác 0 hay bằng 1, nhảy đến mdfy

Mdfy: khi v1 < t1, ta gán số phần tử của biến v0 bằng i + 1

Đặt v1 bằng t1

Test: tang biến i them 1, sử dụng slt và bne để kiểm tra giá trị của i và so sánh với n, nếu i < n, loop sẽ được tiếp tục



Done: xảy ra khi i = n, lúc này nhảy đến continue và xử lý dòng lệnh tiếp theo

Bài 2:

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Chương trình bắt đầu với khai báo các biến trong phần .data. Biến A lưu trữ dãy số cần sắp xếp và biến Aend được sử dụng để xác định phần tử cuối cùng trong dãy số.



Tiếp theo, chương trình bắt đầu ở nhãn main. Ở đây, chương trình sử dụng lệnh la để lưu địa chỉ của phần tử đầu tiên của mảng A vào thanh ghi $a0. Lệnh la cũng được sử dụng để lưu địa chỉ của phần tử cuối cùng của mảng vào thanh ghi $a1. Sau đó, chương trình gọi thủ tục sort để sắp xếp dãy số và khi hoàn thành, sử dụng lệnh li và syscall để kết thúc chương trình.

Thủ tục này sử dụng thuật toán selection sort để sắp xếp dãy số. Nó bắt đầu với việc kiểm tra xem dãy số đã được sắp xếp chưa bằng cách so sánh thanh ghi $a0 và $a1. Nếu chúng bằng nhau, có nghĩa là dãy số chỉ có một phần tử và đã được sắp xếp. Nếu không, chương trình gọi thủ tục max để tìm giá trị lớn nhất trong phần chưa được sắp xếp của mảng và sau đó hoán đổi giá trị lớn nhất với phần tử cuối cùng của phần chưa được sắp xếp. Sau đó, con trỏ $a1 được giảm đi một đơn vị để loại bỏ phần tử cuối cùng khỏi phần chưa được sắp xếp. Thủ tục sắp xếp lặp lại cho đến khi tất cả các phần tử trong dãy đã được sắp xếp.

Hàm max được sử dụng để tìm giá trị lớn nhất trong mảng. Nó có hai tham số đầu vào là con trỏ tới phần tử đầu tiên ($a0) và cuối cùng ($a1) của mảng.

Trong hàm max, ta khởi tạo một số thanh ghi, bao gồm:

$v0 được khởi tạo bằng $a0, chứa con trỏ tới phần tử đầu tiên của mảng.

$v1 được khởi tạo bằng giá trị được lấy từ địa chỉ con trỏ $v0 trỏ đến. Nó chứa giá trị của phần tử đầu tiên của mảng.



Trong hàm loop, các bước được thực hiện như sau:

Kiểm tra xem con trỏ $t0 có trỏ tới phần tử cuối cùng của mảng hay không. Nếu có, quá trình sắp xếp đã hoàn tất và ta trở lại hàm max.

Nếu không phải phần tử cuối cùng của mảng, ta tăng con trỏ $t0 lên 1 đơn vị để trỏ tới phần tử tiếp theo.

Ta tải giá trị của phần tử tiếp theo vào thanh ghi $t1.



Kiểm tra xem giá trị của phần tử tiếp theo có lớn hơn giá trị của phần tử lớn nhất hiện tại hay không. Nếu không phải, ta quay lại bước 2 để tiếp tục kiểm tra với phần tử kế tiếp.

Nếu giá trị của phần tử tiếp theo lớn hơn giá trị của phần tử lớn nhất hiện tại, ta cập nhật con trỏ $v0 trỏ tới phần tử có giá trị lớn nhất hiện tại, và cập nhật giá trị của biến $v1 với giá trị lớn nhất hiện tại. Sau đó, ta quay lại bước 2 để kiểm tra với phần tử kế tiếp.



Sau khi kiểm tra hết tất cả các phần tử trong mảng, hàm loop sẽ trả về con trỏ $v0 trỏ tới phần tử có giá trị lớn nhất và giá trị lớn nhất đó được lưu trữ trong biến $v1.

Bài 3:

Text

Description automatically generated

A picture containing chart

Description automatically generated

Ở phần .data, chúng ta định nghĩa mảng arr gồm 11 số nguyên và một chuỗi trắng " ".

Ở phần .text, chúng ta định nghĩa hàm main.



Đầu tiên, chúng ta gán giá trị địa chỉ của phần tử đầu tiên trong mảng arr vào thanh ghi $s0. Chúng ta khởi tạo các biến $t0, $t1, $s1, $s2, $t2, $t3 để sử dụng trong việc lặp trên mảng.





Sau đó, chúng ta bắt đầu một vòng lặp ngoài với biến $t0 chạy từ 0 đến 10, tương ứng với số phần tử trong mảng. Trong vòng lặp ngoài, chúng ta khởi tạo lại biến $t1 về giá trị 0 và biến $s2 được giảm giá trị đi 1 trong mỗi lần lặp để phù hợp với kích thước của vòng lặp bên trong.

Tiếp theo, chúng ta bắt đầu một vòng lặp trong vòng lặp ngoài để so sánh và đổi chỗ các phần tử liên tiếp của mảng. Trong vòng lặp trong, chúng ta sử dụng thanh ghi $t3 để lặp qua từng phần tử trong mảng arr. Chúng ta sử dụng hướng dẫn lw để tải giá trị của phần tử arr[j] vào thanh ghi $s3, sau đó sử dụng lw để tải giá trị của phần tử tiếp theo arr[j+1] vào thanh ghi $s4. Tiếp theo, chúng ta so sánh giá trị của hai phần tử này. Nếu phần tử $s3 nhỏ hơn phần tử $s4, chúng ta sử dụng một nhánh điều kiện để đổi chỗ hai phần tử. Chúng ta sử dụng hướng dẫn sw để lưu giá trị của phần tử $s3 vào arr[j+1] và giá trị của phần tử $s4 vào arr[j].

Bài 4:

Graphical user interface, text, application

Description automatically generated Text

Description automatically generated

Đầu tiên, mã nguồn khai báo mảng A với 8 phần tử số nguyên, sau đó lưu địa chỉ của phần tử đầu tiên trong mảng A vào thanh ghi $a0.

Ở đoạn mã sau đó, chương trình tiếp tục triển khai thuật toán sắp xếp chèn. Ban đầu, giá trị của biến $t0 được gán bằng 1, đại diện cho chỉ số của phần tử thứ 2 trong mảng A (vì phần tử đầu tiên đã được sắp xếp).

Sau đó, chương trình sử dụng hai vòng lặp lồng nhau để duyệt qua từng phần tử trong mảng và so sánh với các phần tử trước nó để xác định vị trí chính xác của nó trong mảng đã được sắp xếp.

Trong vòng lặp ngoài cùng, biến $t1 được gán bằng giá trị của biến $t0 (i), sau đó chương trình bắt đầu vòng lặp trong cùng (vòng lặp while).

Trong vòng lặp while, địa chỉ của phần tử thứ $j$ và phần tử liền trước nó trong mảng A được tính toán, sau đó hai giá trị này được so sánh để xác định xem chúng có đúng vị trí với nhau hay không.

Nếu giá trị của A[j] nhỏ hơn giá trị của A[j-1], chúng sẽ được hoán đổi với nhau, sau đó biến $j$ sẽ giảm đi 1 để so sánh tiếp tục các phần tử liền trước đó.

Nếu A[j] lớn hơn hoặc bằng A[j-1], biến $j$ sẽ giữ nguyên giá trị của nó và tiếp tục vòng lặp while để so sánh phần tử tiếp theo.

Khi vòng lặp while kết thúc, chương trình tiếp tục vòng lặp for ở bên ngoài để duyệt qua phần tử tiếp theo trong mảng A, cho đến khi tất cả các phần tử trong mảng đã được sắp xếp.

Cuối cùng, chương trình kết thúc và thoát.