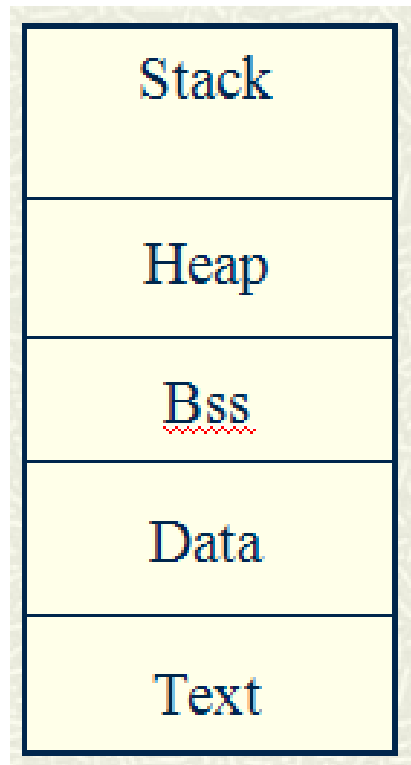


Tập lệnh trình liên kết trong CW10

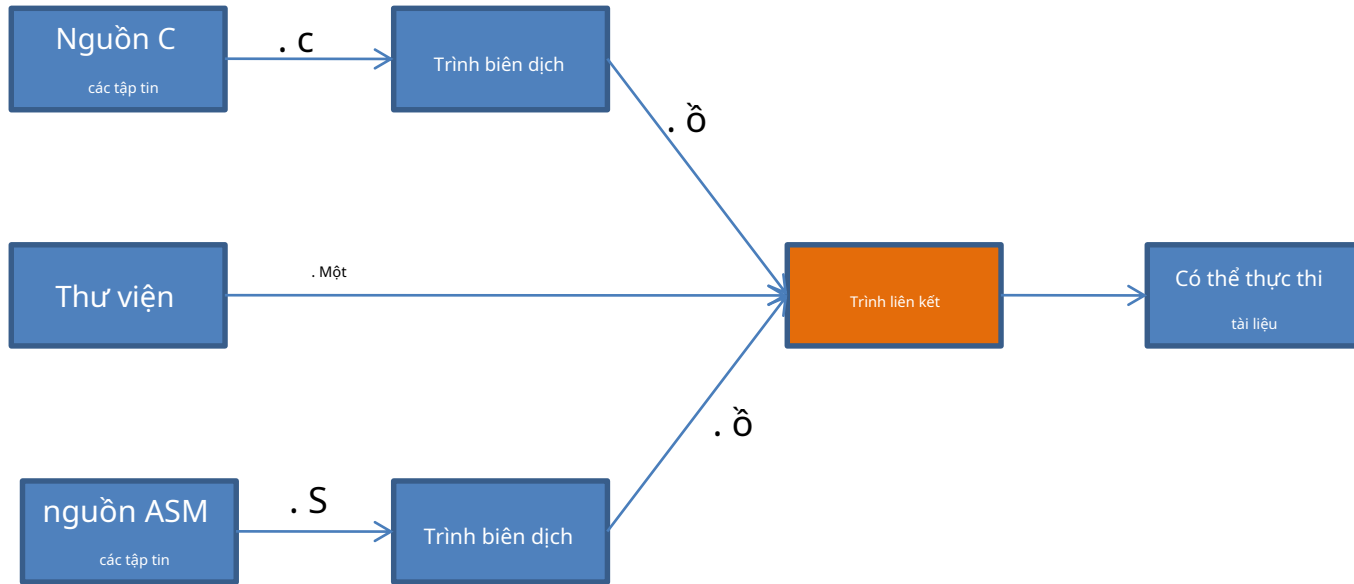
MinhNQ2



- Text- Hướng dẫn chạy chương trình
- Dữ liệu - Các biến toàn cục được khởi tạo.
- Bss – Biến toàn cục chưa được khởi tạo. Chúng được khởi tạo thành số không.
- Heap – Bộ nhớ được trả về khi gọi malloc/new.
- Ngăn xếp - Nó lưu trữ các biến cục bộ và địa chỉ trả về.
- Mỗi phần có các quyền khác nhau: đọc/ghi/thực thi hoặc kết hợp chúng.**



Xây dựng một chương trình



- Các tệp lệnh của trình liên kết bao gồm ba loại phân đoạn, phải theo thứ tự sau:
 - MỘT **ký ức** phân đoạn, bắt đầu bằng **KÝ_ỨC{}** chỉ thị
 - Không bắt buộc **Khép kín** các phân đoạn, bắt đầu bằng **FORCE_ACTIVE{}**, **GIỮ_SECTION{}**, hoặc **REF_INCLUDE{}** chỉ thị
 - MỘT **phần** phân đoạn, bắt đầu bằng **PHẦN{}** chỉ thị

- Sử dụng phân đoạn bộ nhớ để chia bộ nhớ khả dụng thành các phân đoạn.
- Các (**RWX**) bao gồm các cờ cho phép truy cập ELF: R = đọc, W = ghi hoặc X= thực thi.
- **NGUỒN GỐC** chỉ định địa chỉ bắt đầu của phân đoạn bộ nhớ, hoặc là địa chỉ bộ nhớ thực hoặc thông qua **SAU ĐÓ** từ khóa, tên của phân đoạn trước đó.
- **CHIỀU DÀI** chỉ định kích thước của phân đoạn bộ nhớ. Giá trị 0 có nghĩa là độ dài không giới hạn.

- Các phân đoạn đóng cho phép bạn làm cho các biểu tượng không bị loại bỏ.
- ***FORCE_ACTIVE***-Sử dụng lệnh này để làm cho trình liên kết bao gồm một ký hiệu mà nếu không thì nó sẽ không bao gồm.
- ***GIỮ_SECTION***-Sử dụng lệnh này để giữ một phần trong liên kết, đặc biệt là phần do người dùng xác định.
- ***REF_INCLUDE***-Sử dụng lệnh này để giữ một phần trong liên kết, miễn là có tham chiếu đến tệp chứa phần đó. Đây là một cách hữu ích để bao gồm số phiên bản.

- Sử dụng phân đoạn các phần để xác định nội dung của các phần bộ nhớ và để xác định bất kỳ ký hiệu chung nào mà bạn muốn sử dụng trong tệp đầu ra của mình.
- Định dạng

```
PHẦN {  
    tên phần : [AT (loadAddress)]  
    {  
        Nội dung  
    } > tên phân đoạn  
}
```

} phần_spec

- **tên phần** : Tên phần đầu ra
- **AT (địa chỉ tải)** : Tùy chọn chỉ định địa chỉ tải của phần. Giá trị mặc định là địa chỉ di dời.
- **Nội dung** : Các câu lệnh gán giá trị cho một ký hiệu hoặc chỉ định vị trí của phần, bao gồm cả các phần đầu vào.
- **tên phân khúc** : Đích phân đoạn bộ nhớ được xác định trước cho nội dung của phần. Hai biến thể là:
 - > SegName: đặt nội dung của phần ở đầu phân đoạn bộ nhớ.
 - > > SegName: nối nội dung của phần vào cuối phân đoạn bộ nhớ.

- Biến, biểu thức và tích phân
- Toán tử số học, nhận xét
- Căn chỉnh
- Chỉ định tập tin và chức năng
- Xếp chồng và xếp đồng
- Sao chép ROM-RAM

- Tất cả tên ký hiệu phải bắt đầu bằng ký tự gạch dưới (_). Các ký tự khác có thể là chữ cái, chữ số hoặc dấu gạch dưới.
- ***_symbolicname = some_biểu thức;***
- Có 2 kiểu biểu đạt
 - ***biểu thức tuyệt đối***—biểu tượng chứa giá trị mà nó sẽ có trong tệp đầu ra.
 - ***Biểu thức có thể định vị lại***—biểu thức giá trị là một phần bù cố định từ phần cơ sở của một phần
- Cú pháp LCF cho các biểu thức rất giống với cú pháp của ngôn ngữ lập trình C
 - `_số_thập phân = 123245;`
 - `_hex_number = 0x999999FF;`

Quyền ưu tiên	Toán tử
1	- ~ !
2	* / %
3	+ -
4	> > <<
5	== != > < <= >=
6	&
7	
số 8	&&
9	

- Sử dụng ký tự sắc nét “#” cho một dòng bình luận
 - *#đây là một bình luận một dòng*
- Sử dụng dấu gạch chéo và dấu hoa thị “/*” cho bình luận nhiều dòng
 - */*đây là*
 - *bình luận nhiều dòng */*
- Sử dụng dấu gạch chéo kép “//” cho nhận xét một phần dòng
 - *//đây là nhận xét một phần dòng*

Chỉ định tập tin và chức năng

- Chỉ định tệp: Xác định nội dung của một phân đoạn phần bao gồm việc chỉ định tệp nguồn của từng phần.
 - Cách 1: liệt kê các tập tin
 - Phương pháp 2: sử dụng ký tự đại diện dấu hoa thị (*), đại diện cho tên của mọi tệp trong dự án của bạn.
- Chỉ định tập tin: Để kiểm soát chính xác vị trí chức năng trong một phần, hãy sử dụng **SỰ VẬT** từ khóa.
 - *PHẦN {*
 - *. phần_chương trình :*
 - *{*
 - *SỰ VẬT (beta, C chính)*
 - *SỰ VẬT (alpha, C chính)*
 - ** (.chữ)*
 - *} > GỐC*

- Dành không gian cho ngăn xếp và vùng nhớ heap yêu cầu một số phép tính số học để đặt giá trị ký hiệu được sử dụng trong thời gian chạy.
- Hoạt động thiết lập ngăn xếp
 - **`_stack_address = __END_BSS;`**
 - **`_stack_address = _stack_address & ~7; /*căn chỉnh đỉnh ngăn xếp 8 */`**
 - **`__SP_INIT = _stack_address + 0x4000; /*đặt ngăn xếp thành 16KB*/`**
- Hoạt động thiết lập vùng heap
 - **`__heap_addr = __SP_INIT; /* heap phát triển ngăn xếp đối diện */`**
 - **`__heap_size = 0x50000; /* kích thước vùng heap được đặt thành 500KB */`**

- Thông thường dữ liệu hoặc mã của chương trình nằm trong ROM sẽ được sao chép vào RAM khi chạy. Để chỉ ra dữ liệu hoặc mã đó, hãy sử dụng LCF để gán cho nó hai địa chỉ:
 - Đoạn bộ nhớ chỉ định vị trí dự định trong RAM
 - Phân đoạn phần xác định vị trí thường trú trong ROM, thông qua tham số AT (địa chỉ) của nó
- Để thực hiện chương trình sao chép phần từ ROM sang RAM, bảng sao chép phải cung cấp thông tin mà chương trình cần khi chạy. Bảng sao chép này, có ký hiệu __S_romp xác định, chứa một chuỗi gồm ba giá trị từ cho mỗi mục nhập:
 - Địa chỉ bắt đầu của ROM
 - Địa chỉ bắt đầu RAM
 - Kích cỡ

Lệnh, Chỉ thị và Từ khoá - 1

<u>. (location counter)</u>	<u>ADDR</u>	<u>ALIGN</u>
<u>ALIGNALL</u>	<u>EXCEPTION</u>	<u>FORCE_ACTIVE</u>
<u>INCLUDE</u>	<u>KEEP_SECTION</u>	<u>MEMORY</u>
<u>OBJECT</u>	<u>REF_INCLUDE</u>	<u>SECTIONS</u>
<u>SIZEOF</u>	<u>SIZEOF_ROM</u>	<u>WRITEB</u>
<u>WRITEH</u>	<u>WRITEW</u>	<u>WRITES0COMMENT</u>
<u>ZERO_FILL_UNINITIALIZED</u>		

- **(bộ đếm vị trí):** Biểu thị vị trí đầu ra hiện tại

. dữ liệu :

{

**. (dữ liệu)*

**. (bss)*

**. (CHUNG)*

__bắt đầu = .;

. = __bắt đầu + 0x10000; __kết

thức = .;

} > DỮ LIỆU

- **ĐỊA CHỈ:** Trả về địa chỉ của phần được đặt tên hoặc phân đoạn bộ nhớ

ĐỊA CHỈ (tên phần / tên phân đoạn)

- phầnName: Mã định danh cho một phần tệp.
- SegName: Mã định danh cho phân đoạn bộ nhớ

- **CĂN CHỈNH:**Trả về giá trị bộ đếm vị trí, được căn chỉnh trên một ranh giới được chỉ định.

CÁNH(*alignValue*)

AlignValue: Công cụ xác định ranh giới căn chỉnh; phải là lũy thừa của hai.

- **CÂN CHỈNH:**Buộc căn chỉnh tối thiểu tất cả các đối tượng trong phân đoạn hiện tại theo giá trị đã chỉ định.

ALIGNALL(*alignValue*);

AlignValue: Công cụ xác định giá trị căn chỉnh; phải là lũy thừa của hai.

- **FORCE_ACTIVE**: Bắt đầu một phân đoạn đóng LCF tùy chọn để chỉ định các ký hiệu mà trình liên kết không nên ngắt.

FORCE_ACTIVE{ ký hiệu[, ký hiệu] }

ký hiệu: Bất kỳ ký hiệu nào được xác định.

- **BAO GỒM**: Bao gồm một tệp nhị phân được chỉ định trong tệp đầu ra.

BAO GỒM tên tệp

filename: Tên của tệp nhị phân. Đường dẫn của tệp nhị phân cần được chỉ định làm đối số dòng lệnh của trình liên kết.

- **GIỮ_SECTION**: Bắt đầu một phân đoạn đóng LCF tùy chọn để chỉ định các phần mà trình liên kết không nên loại bỏ dải chết.

KEEP_SECTION{sectionType[, partType] }

phầnType: Mã định danh cho bất kỳ phần nào do người dùng xác định hoặc xác định trước.

- **KÝ Ứ:** Bắt đầu phân đoạn bộ nhớ LCF, xác định các phân đoạn của bộ nhớ đích.

BỘ NHỚ {bộ nhớ_spec[, bộ nhớ_spec] }

bộ nhớ_spec:

SegName (accessFlags): ORIGIN = địa chỉ, LENGTH = chiều dài [> fileName]

- **tên phân khúc:** Tên cho một phân đoạn mới của bộ nhớ đích.
- **cờ truy cập:** Cờ cho phép truy cập ELF - R = đọc, W = ghi hoặc X = thực thi.
- **Địa chỉ:** Địa chỉ bộ nhớ hoặc lệnh SAU
- **Chiều dài:** Kích thước của phân đoạn bộ nhớ mới
- **Tên tập tin:** Đích tập nhị phân tùy chọn

- **KÍCH THƯỚC:** Trả về kích thước (tính bằng byte) của phân đoạn hoặc phần đã chỉ định.

SIZEOF(tên phân đoạn / tên phần)

- tên phân đoạn: Tên của một phân đoạn

- **SIZEOF_ROM:** Trả về kích thước (tính bằng byte) của phân đoạn hoặc phần đã chỉ định.

SIZEOF(tên phân đoạn / tên phần)

- tên phân đoạn: Tên của phân đoạn ROM

- **VIẾT, VIẾT, VIẾT:** Chèn một byte dữ liệu vào địa chỉ hiện tại của một phần.

WRITEx (biểu thức);

- sự biểu lộ: Bất kỳ biểu thức nào trả về một giá trị trong phạm vi

- **VIẾTBÌNH LUẬN:**Chèn bản ghi nhận xét S0 vào tệp bản ghi S.

WRITES0COMMENT "bình luận"

- bình luận:văn bản bình luận

- **ZERO_FILL_UNINITIALIZED:**Buộc trình liên kết đưa dữ liệu bằng 0 vào tệp nhị phân cho các biến chưa được khởi tạo. Lệnh này phải nằm giữa lệnh MEMORY và SECTIONS; đặt nó ở bất cứ nơi nào khác sẽ là một lỗi cú pháp.

ZERO_FILL_UNINITIALIZED

- **Định dạng**

pragma định nghĩa_section sname ".istr" [.ustr] [.rostr] [addrmode] [accmode]

*# bắt đầu tên phần pragma /**

*Mã số */*

pragma phần sname end __declspec(section

"sname ") <prototype>;

- **Thông số**

- tên của: Mã định danh cho các tham chiếu nguồn tới phần do người dùng xác định này.
- istr: Chuỗi tên phần cho dữ liệu khởi tạo được gán cho phần này
- sử dụng: Tùy chọn: Tên phần ELF cho dữ liệu chưa được khởi tạo được gán cho phần này.
- danh sách: Tùy chọn: Tên phần ELF cho dữ liệu chỉ đọc được gán cho phần này.
- mã bổ sung: Tùy chọn: tham số tùy chọn cho biết cách trình liên kết xử lý phần đó.
- tài khoản: quyền truy cập trong phần này

Cảm ơn

Hỏi đáp

