TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA MẠNG MÁY TÍNH & TRUYỀN THÔNG

ĐỀ THI THỬ

Tên môn học: Lập trình hệ thống

Thời gian làm bài: 45 phút

Giám thị 1	Giám thị 2

Họ, tên SV: Phan Hoàng Dũng
Mã SV: 24N2034
STT: .....

(Thí sinh không được sử dụng tài liệu)

Mã đề thi 001

Điểm (số):	Giám khảo 1	Giám khảo 2	Số phách
Điểm (chữ):			

MÃ ĐỀ	SỐ BÁO DANH	ĐÁP ÁN TRẮC NGHIỆM
☐☐☐	☐☐☐☐☐☐	
0 ○○○	0 ○○○○○○	1 Ⓐ○○○ 11 ○○○○ 21 ○Ⓑ○○ 31 ○○○○
1 ○○○	1 ○○○○○○	2 Ⓐ○○○ 12 ○○○○ 22 ○Ⓑ○○ 32 ○○○○
2 ○○○	2 ○○○○○○	3 ○Ⓑ○○ 13 ○○○○ 23 ○Ⓑ○○ 33 ○○○○
3 ○○○	3 ○○○○○○	4 Ⓐ○○○ 14 ○○○○ 24 ○Ⓑ○○ 34 ○○○○
4 ○○○	4 ○○○○○○	5 Ⓐ○○○ 15 ○○○○ 25 ○Ⓑ○○ 35 ○○○○
5 ○○○	5 ○○○○○○	6 Ⓐ○○○ 16 ○○○○ 26 ○Ⓑ○○ 36 ○○○○
6 ○○○	6 ○○○○○○	7 Ⓐ○○○ 17 ○○○○ 27 ○Ⓑ○○ 37 ○○○○
7 ○○○	7 ○○○○○○	8 Ⓐ○○○ 18 ○○○○ 28 ○Ⓑ○○ 38 ○○○○
8 ○○○	8 ○○○○○○	9 Ⓐ○○○ 19 ○○○○ 29 ○Ⓑ○○ 39 ○○○○
9 ○○○	9 ○○○○○○	10 Ⓐ○○○ 20 ○○○○ 30 ○Ⓑ○○ 40 ○○○○

Thí sinh lưu ý :
- Giữ cho phiếu phẳng, không bôi bẩn, làm rách, không tẩy xóa, để máy chấm.
- Tô **kín**, tô **đậm** các ô tròn tương ứng với mã Đề thi, Số báo danh và đáp án đúng cho Phần trắc nghiệm.
- Không được ghi đè, tô đè lên các ô vuông đen, để máy định vị chính xác
- **Chỉ chọn một đáp án** (Không bôi mờ các đáp án khác để máy chấm chính xác)
- Số báo danh: 6 chữ số - phiên bản rút gọn của MSSV. Ví dụ: **23**52**0560** → 230560
- Mã đề: 3 chữ số - ghi và tô đúng và đủ

Câu 1. Cho mảng **short a[5]**, chọn lệnh đúng để lấy giá trị của phần tử a[i], biết địa chỉ của a lưu ở %eax, chỉ số i lưu ở %edx?

A. `movw (%eax, %edx, 2), %bx`
B. `movl (%eax, %edx, 2), %ebx`
C. `movw (%edx, %eax, 2), %bx`
D. `movl (%edx, %eax, 2), %ebx`

Câu 2. Giá trị trả về của một hàm trong hệ thống 64 bit được lưu ở đâu?

A. Thanh ghi **B.** Stack **C.** Section .data **D.** Khác

Câu 3. (IA32) Trong stack frame của 1 hàm, vị trí %ebp trỏ đến luôn cố định, đó là vị trí nào?

A. Ô nhớ lưu địa chỉ trả về của hàm
B. Ô nhớ lưu %ebp của hàm mẹ
C. Ô nhớ lưu tham số thứ nhất
D. Ô nhớ lưu giá trị trả về của hàm

Câu 4. Đối tượng nào sau đây KHÔNG phải là symbol đối với các linker?

A. Biến toàn cục
B. Biến cục bộ có từ khóa static
C. Hàm có từ khóa static
D. Không có câu nào đúng

Câu 5. Cho 1 hàm trong kiến trúc 64 bit có 8 tham số, các tham số này sẽ được truyền như thế nào dưới hệ thống?

A. Không hỗ trợ truyền tham số
B. Toàn bộ tham số được đưa vào stack
C. Toàn bộ tham số truyền qua thanh ghi
D. Khác

Câu 6. Ảnh hưởng của lệnh `push %ebp` trong hệ thống 32 bit lên 2 thanh ghi `%ebp` và `%esp`?

A. 2 thanh ghi không thay đổi
B. %esp bị giảm xuống 4
C. %ebp không thay đổi
D. Câu B và C đúng

Câu 7. So sánh stack frame của hàm mẹ và hàm con, câu nào đúng?

A. Stack frame hàm mẹ được cấp phát trước và thu hồi trước stack frame hàm con
B. Stack frame hàm mẹ được cấp phát sau và thu hồi sau stack frame hàm con
C. Hàm mẹ và hàm con có thể truy xuất biến cục bộ trong stack frame của nhau
D. Stack frame hàm mẹ nằm ở địa chỉ cao hơn so với stack frame hàm con

Câu 8. Cho mảng `short c[5][5]`, kích thước của mảng là bao nhiêu byte?

A. 10 bytes **B.** 25 bytes **C.** 50 bytes **D.** 100 bytes

Câu 9. Cho **%esp = 0x1028**, tính giá trị của **%esp** sau khi thực hiện đoạn lệnh sau:

```
1  push %ebp
2  subl $20, %esp
```

A. 0x1004 **B.** 0x1010 **C.** 0x1014 **D.** 0x1008

Sử dụng dữ kiện sau để trả lời các câu hỏi 10, 11, 12, 13:

Cho struct rec trong Linux 32 bit: `struct rec { float a; double b; char c; }`

Câu 10. Yêu cầu căn chỉnh chung K của **struct rec** là bao nhiêu?

A. K = 1 **B.** K = 4 **C.** K = 8 **D.** K = 16

Câu 11. Nhận định nào đúng về địa chỉ của các thành phần trong **struct rec**?

A. Thành phần a có địa chỉ cao nhất
B. Thành phần c có địa chỉ cao nhất
C. Cả 3 thành phần đều có chung địa chỉ
D. Không có đáp án đúng

Câu 12. Tổng kích thước của **struct rec** là bao nhiêu (có tính alignment)?

A. 13 bytes **B.** 16 bytes **C.** 20 bytes **D.** Khác

Câu 13. Trong trường hợp có và không có alignment, chênh lệch kích thước của struct rec là?

A. 3 bytes **B.** 4 bytes **C.** 5 bytes **D.** Khác

Sử dụng dữ kiện sau để trả lời các câu hỏi 14, 15:

Cho union rec trong Linux 32 bit, bỏ qua alignment.

`union rec { float a; char c; double b; }`

Câu 14. Kích thước của union là bao nhiêu?

A. 4 bytes **B.** 8 bytes **C.** 9 bytes **D.** 13 bytes

Câu 15. Giả sử địa chỉ của union đang lưu trong %eax, lệnh assembly nào có thể dùng để truy xuất thành phần c của union trên?

A. `movl (%eax), %ebx`
B. `movb 4(%eax), %bl`
C. `movl 4(%eax), %ebx`
D. Khác

Câu 16. Section chứa biến cục bộ có static chưa khởi tạo giá trị của 1 hàm trong file thực thi?

A. Không nằm trong section nào
B. Section .data
C. Section .bss
D. Section .text

Câu 17. Chọn câu đúng khi nói về Linker (ld)?

A. Linker nhận đầu vào là các file source .c

B. Linker cần đầu vào tối thiểu là 2 file để thực hiện liên kết

C. Linker làm việc trên các hàm và biến cục bộ

D. Linker không thể liên kết 2 file source có định nghĩa đầy đủ 2 hàm trùng tên

Câu 18. Khi khai thác buffer overflow trong 1 hàm có lỗ hổng bằng 1 chuỗi input, ghi đè tùy ý lên ô nhớ nào trong stack frame của hàm có thể gây ra lỗi segmentation fault?

A. Ô nhớ chứa giá trị trả về

B. Ô nhớ được trỏ đến bởi %ebp

C. Ô nhớ chứa biến toàn cục

D. Ô nhớ nằm ở địa chỉ thấp hơn chuỗi input

Câu 19. Cho đoạn mã assembly bên dưới. Giá trị của %eax sau đoạn lệnh là bao nhiêu?

```
1  movl  $0x120, %eax
2  movb  $0x0, %ah
3  incl  %eax
```

A. 0x121 **B.** 121 **C.** 0x101 **D.** 0x21

Câu 20. Mảng nào dưới đây có tổng kích thước thay đổi trong trường hợp khai báo trên hệ thống 32bit và 64 bit?

A. int a[5][5] **B.** long c[3][3] **C.** short b[2][3] **D.** char d[4][2]

Câu 21. Cho struct như bên dưới trên hệ thống Linux 32 bit có alignment. Hỏi trong trường hợp tối ưu hóa để tiết kiệm không gian lưu trữ, kích thước struct có thể giảm bao nhiêu byte?

```
struct ex { char a; char* b; double c; short d};
```

A. 2 bytes **B.** 3 bytes **C.** 4 bytes **D.** Khác

Câu 22. Trong symbol table của mô-đun *m.o* có 1 dòng symbol có tên **sym1** với thông tin sau:

Num	Value	Size	Type	Bind	Ndx	Name
..	..	..	..	GLOBAL	UND	sym1

Chọn câu **ĐÚNG**?

A. sym1 không phải là symbol

B. sym1 là 1 local symbol

C. sym1 được định nghĩa đầy đủ trong *m.o*

D. sym1 được định nghĩa đầy đủ ở một file khác *m.o*

Câu 23. Cho đoạn code C và mã assembly tương ứng:

```
1.  int func(){
2.      char str[5];
3.      gets(str);
4.  }
```

```
1.  <func>:
2.      pushl   %ebp
3.      movl    %esp,%ebp
4.      subl    $0x18,%esp
5.      leal    -0xd(%ebp),%eax
6.      pushl   %eax
7.      call    80482e0 <gets@plt>
```

Một sinh viên nhập tay từng ký tự từ bàn phím để đoán trường hợp lỗi chương trình, vậy chuỗi str nào sau đây đã có thể gây ra lỗi Segmentation fault?

A. 4 ký tự 'A' **B.** 5 ký tự 'A' **C.** 13 ký tự 'A' **D.** Không có chuỗi nào

Sử dụng dữ kiện sau để trả lời các câu hỏi 24, 25:

Cho mảng **int B[N][5]** với N chưa biết, có tồn tại thành phần B[2][3].

Câu 24. Tổng kích thước của B có thể là bao nhiêu?

A. 40 bytes **B.** 60 bytes **C.** 50 bytes **D.** 70 bytes

Câu 25. Cho địa chỉ bắt đầu là 0x1010. Cho biết địa chỉ của thành phần B[2][3]?

A. 0x1016 **B.** 0x1028 **C.** 0x1044 **D.** Không đủ dữ kiện

-- Chúc các bạn ôn thi tốt --