HSGS Sec Day 04

Syscall, sections, NASM, Linux

Agenda

1. Quiz & Homework
2. Syscall
3. Sections
4. NASM
5. Linux
6. GDB

Syscall

- Chương trình (userspace) cần giao tiếp với hệ điều hành để có thể thực hiện các tác vụ phức tạp
- Instruction mới: SYSCALL
- Có thể hiểu là 1 hàm do hệ điều hành cung cấp
- Arguments
  - RAX: mã số hàm (selector): Linux System Call Table for x86 64 · Ryan A. Chapman
  - RDI: argument thứ 1
  - RSI: argument thứ 2
  - RDX: argument thứ 3
  - R10: argument thứ 4
  - R8: argument thứ 5
- Return value: RAX: < 0: fail; >=0: success

Syscall - exit

- int sys_exit(int status)
- RAX: 60 (Linux)
- Kết thúc chương trình, với code status

Syscall - read

- int sys_read(unsigned int fd, char *buf, size_t count)
- RAX: 0 (Linux)
- fd: file descriptor: 1 số dùng để miêu tả 1 file
  - 0: stdin
  - 1: stdout
  - 2: stderr
- Đọc count bytes từ file fd vào buf
- Return (RAX): số bytes đọc được; < 0: fail

Syscall - write

- int sys_write(unsigned int fd, char *buf, size_t count)
- RAX: 1 (Linux)
- fd: file descriptor: 1 số dùng để miêu tả 1 file
  - 0: stdin
  - 1: stdout
  - 2: stderr
- Viết count bytes từ buf vào file fd
- Return (RAX): số bytes đã viết được; < 0: fail

Sections

- 1 file executable có nhiều sections, trong số đó bao gồm
  - .text (R/X): Chứa các instructions để thực hiện
  - .data (R/W): Chứa các biến global có giá trị được định nghĩa
  - .rodata (R): global consts
  - .bss (R/W): biến global được khởi tạo giá trị = 0

NASM

- Assembly compiler
- nasm -f elf64 -g -o a.o a.s: Biên dịch file .s thành file object
- ld -static -o a a.o: link file .o thành chương trình

```
1   section .text
2   global _start
3   %define SYS_EXIT 60
4   _start:
5       push rbp
6       mov rbp, rsp
7
8       xor rdi, rdi
9       mov rax, SYS_EXIT
10      syscall
11      hlt
12      leave
13      ret
14
```

NASM

- section sectname: phần ở dưới nó thuộc section sectname
- global _start: export label _start để có thể định nghĩa vị trí code đầu tiên được thực hiện
- label .L1 có tên đầy đủ là _start.L1
  - Không chấm: hàm
  - Chấm: nhánh của hàm
-

```
1   section .text
2   %define SYS_READ 0
3   %define SYS_WRITE 1
4   %define SYS_EXIT 60
5   global _start
6   _start:
7       mov rdi, 1              ; fd
8       mov rsi, s              ; buf
9       mov rdx, s.end - s      ; length
10      mov rax, SYS_WRITE
11      syscall
12      xor rdi, rdi            ; status
13      mov rax, SYS_EXIT
14      syscall
15      hlt
16
17  section .data
18  a:
19      db 0 ; 1 byte
20  b:
21      dq 1337 ; 8 byte
22  s:
23      db "Hello World", 0x0
24  .end:
25
```

Linux

- Giao diện dòng lệnh
- Các câu lệnh thường sử dụng
  - ls: liệt kê các file/folder trong thư mục hiện tại
    - ls folder: liệt kê các file/folder trong thư mục
  - cd folder: chuyển thư mục hiện tại sang folder
  - Chú ý: folder là .. (2 chấm) nếu muốn chỉ tới thư mục phía trước
  - rm file: xoá file (xóa luôn không phục hồi được)
    - rm a: xoá file a trong thư mục hiện tại
    - rm ../b: xoá file b trong thư mục parent của thư mục hiện tại
  - nano file: chỉnh sửa file, tạo file mới nếu chưa tồn tại
  - passwd: đổi password

Linux

- SSH Client:
  - Linux/MacOS: terminal ssh command
    - ssh username@address (nếu port là 22)
    - ssh username@address -P port
  - Secure Shell – Chrome Web Store (google.com)
- SSH Server
  - Address: TBA
  - Port: TBA

GDB (GNU Debugger)

- gdb executable: load binary
- quit: thoát gdb
- help command: hướng dẫn sử dụng command
- run (hoặc r): chạy chương trình
- Ctrl+C: ngắt chương trình
- i r (info registers): đọc các thanh ghi
- x address/$register/expression/label: in giá trị tại địa chỉ
  - Check *help x* để biết thêm các format tồn tại
- b label/address: ngắt chương trình khi chạy đến label/address
  - Instruction int3 cũng có tác dụng tương tự
- delete x: xoá breakpoint x
- c: tiếp tục chương trình sau khi đã ngắt

GDB

- Si: step instruction: thực hiện 1 instruction
- set disassembly-flavor intel: sử dụng intel assembly syntax
- list(l): in source tại instruction hiện tại (nếu có -g và có source file)
- define hook-stop: các câu lệnh được thực hiện khi chương trình bị ngắt
    - E.g: info registers; list; x/5i $pc (in 5 chỉ dẫn tiếp theo)
    - end

Mở rộng: GDB & C/C++

- Compile C/C++: gcc/g++: cú pháp gần tương tự nasm
- Compile với -g để có thêm thông tin
- gdb> list: in source
- gdb> info locals: in biến trong hàm
- gdb> p variable: in biến
- gdb> b function/line number

Homework

- Codefun:
- Write: P33014
- Read & Write: P025, P032

Resources

- Linux System Call Table for x86 64 · Ryan A. Chapman
- explainshell.com

Phụ lục: Đề homework

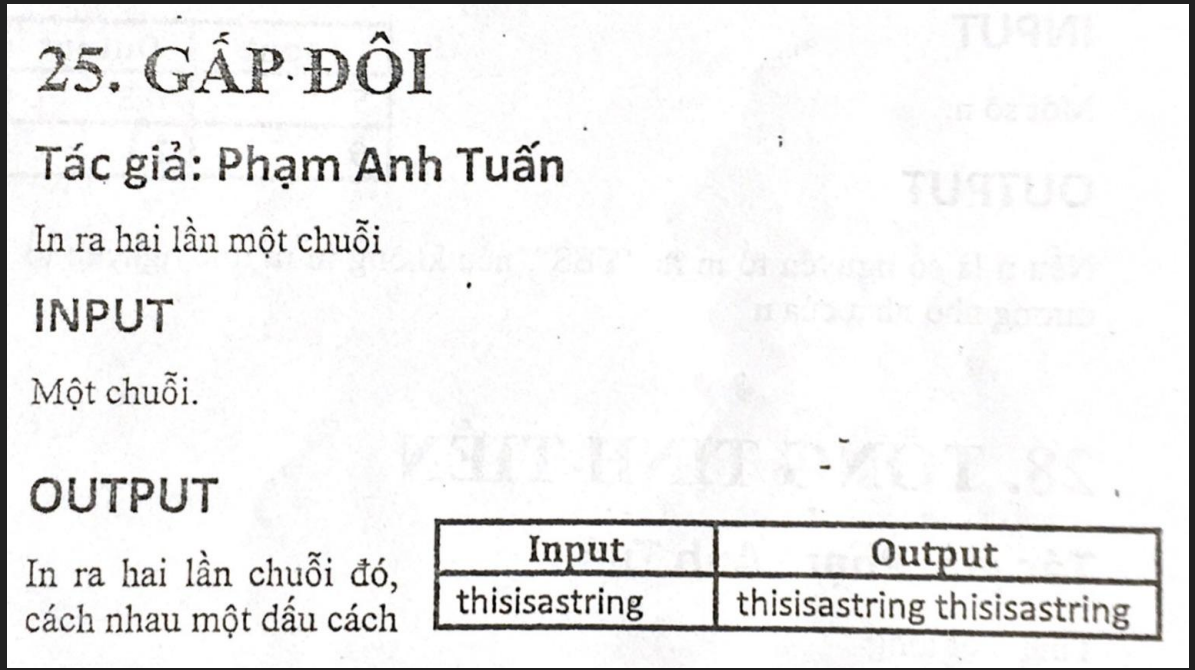
-

P33014

- In ra 100 dòng “Chau Bac Ho”

P025

- Gợi ý: có thể check các kí tự ‘ ‘ (0x20); ‘\n’ (0xa); 0x0; read fail để tìm bắt đầu và kết thúc của xâu.



25. GẤP ĐÔI

Tác giả: Phạm Anh Tuấn

In ra hai lần một chuỗi

INPUT

Một chuỗi.

OUTPUT

In ra hai lần chuỗi đó, cách nhau một dấu cách

Input	Output
thisisastring	thisisastring thisisastring

P032

32. CHUỖI THẦN KỲ

Tác giả: Phạm Anh Tuấn

Nhập vào một chuỗi. In ra màn hình theo mẫu sau:

Input	Output
string	s
	st
	str
	stri
	strin
	string
	string
	strin
	stri
	str
	st
	s

33. PHÉP CHIA

Tác giả: Phạm Anh Tuấn