Sistemas Operacionais

Alisson Linhares

CAMPINAS,

2021/1

Chamadas de sistema

Olá mundo usando Assembly x86_64

```
global _start
section .text
_start:
        mov rax, 1          ; Chamando a syscall write.
        mov rdi, 1          ; 0 = stdin, 1 = stdout, 2 = stderror
        mov rsi, msg        ; rsi contém o ponteiro para a string msg (&msg)
        mov rdx, 10         ; Total de bytes impressos pela syscall write
        syscall             ; Saltando para rotina do SO

        mov rax, 60         ; 60 é o syscall exit
        xor rdi, rdi        ; mesmo que "mov rdi, 0" -> exit code 0
        syscall             ; Saltando para rotina do SO

section .data
        msg: db "Ola mundo", 10     ; 10 é ASCII newline
```

Olá mundo usando Assembly x86_64

- Para compilar e rodar a aplicação no linux, use os comandos abaixo:
  - **nasm -felf64 hello.s**
  - **ld hello.o**
  - **./a.out**

- Lista completa de syscalls + links para código fonte:
  - https://filippo.io/linux-syscall-table/

- Registradores usados em cada syscall:
  - https://chromium.googlesource.com/chromiumos/docs/+/master/constants/syscalls.md

- Resumo dos manuais da Intel:
  - https://www.felixcloutier.com/x86/

Calling Conventions

arch	syscall NR	return	arg0	arg1	arg2	arg3	arg4	arg5
arm	r7	r0	r0	r1	r2	r3	r4	r5
arm64	x8	x0	x0	x1	x2	x3	x4	x5
x86	eax	eax	ebx	ecx	edx	esi	edi	ebp
x86_64	rax	rax	rdi	rsi	rdx	r10	r8	r9

NR	syscall name	references	%rax	arg0 (%rdi)	arg1 (%rsi)	arg2 (%rdx)	arg3 (%r10)	arg4 (%r8)	arg5 (%r9)
0	read	man/ cs/	0x00	unsigned int fd	char *buf	size_t count	-	-	-
1	write	man/ cs/	0x01	unsigned int fd	const char *buf	size_t count	-	-	-
2	open	man/ cs/	0x02	const char *filename	int flags	umode_t mode	-	-	-
3	close	man/ cs/	0x03	unsigned int fd	-	-	-	-	-

Chamando rotinas Assembly em C/C++

```
; <arquivo sum3.s>
global sum3
section .text

sum3:                   ; rax sum3(rdi, rsi, rdx)
    mov rax, rdi        ; rax   = rdi;
    add rax, rsi        ; rax += rsi;
    add rax, rdx        ; rax += rdx;
    ret                   ; return rax;
```

Chamando rotinas Assembly em C/C++

```c
// <arquivo main.c>
#include <stdio.h>

long sum3(long, long, long);

int main() {
    printf("%ld\n", sum3(0, 2, 4));
    printf("%ld\n", sum3(1, 3, 5));
    printf("%ld\n", sum3(1, -1, 1));
    printf("%ld\n", sum3(-1, 1, -1));
    return 0;
}
```

Chamando rotinas Assembly em C/C++

- Para compilar e rodar a aplicação no Linux, use os comandos abaixo:
  - **nasm -felf64 sum3.s**
  - **gcc sum3.o main.c**
  - **./a.out**

LAB 02

Lab 2

- **Q1.** Abra o terminal do Linux e digite os comandos abaixo:

a) cd ~
mkdir lab02
cd lab02
mkdir nova_pasta
cd nova_pasta
pwd
cd ~
pwd

b) cd ~/lab02/nova_pasta
pwd
cd ..
pwd
cd ..
pwd

c) cd ~/lab02/nova_pasta
pwd
cd -
pwd
cd --
pwd

d) cd /
pwd
ls
cd ~

e) cd /proc/
pwd
cat /proc/meminfo
cat /proc/cpuinfo
cd ~
pwd

f) cd ~/lab02
cat /proc/meminfo /proc/cpuinfo > system_info
grep cpu system_info
grep Mem system_info
grep -R cpu
grep -R Mem

g) grep Mem /proc/meminfo
grep cpu /proc/cpuinfo
cat /proc/meminfo | grep Mem
cat /proc/cpuinfo | grep cpu

Q1.A: Explique com suas palavras o que os comandos "**cd -**", "**cd ~**", "**cd /**", "**cd ..**", "**grep**", "**grep -R**" e "**|**" fazem.
Q1.B: Explique com suas palavras o que você viu dentro dos diretórios "**/**" e "**/proc/**".

- **Q2:** Com base no código do **slide 3**, escreva um programa em assembly que imprima o seu nome e o seu RA.
  - **Q2.A:** Rode o comando "**objdump -D a.out**" e compare o assembly que você escreveu, com o assembly gerado pelo linker. Documente no lab02.txt o que você descobriu.
  - **Q2.B:** Resolva o mesmo problema, usando C/C++ e compile normalmente com os comandos "**g++ prog.cpp**" ou "**gcc prog.c**". Rode o comando "**objdump -D a.out**" novamente e compare o resultado do programa gerado pelo compilador com o do programa que você escreveu em assembly. Documente o que você descobriu no lab02.txt.
- **Q3:** Com base nos **slides 6, 7 e 8**. Escreva a sua própria rotina de print em assembly. Escreva um programa em C/C++ e invoque a sua rotina, imprimindo o seu RA e o seu nome.