



## Chamadas do Sistema

Universidade Federal de Uberlândia  
Faculdade de Computação  
Prof. Dr. rer. nat. Daniel D. Abdala

## Na Aula Anterior ...

- Instruções de controle de fluxo;
- Codificando fluxo em Assembly:
  - If-then
  - If-then-else
  - Switch-case
- Codificando repetições em Assembly:
  - while() / do while()
  - for()
- Instruções de acesso a memória;
- Palavras Alinhadas e desalinhas;
- Instruções de transferência de dados entre o Processador e o Coprocessador C1;
- Arranjos;
- Estruturas

2

## Nesta Aula

- Introdução;
- Saída de Dados;
- Entrada de Dados;
- Outros Serviços do Sistema;
- Término do programa;

3

## Introdução

- O processador apenas é capaz de executar as instruções exatamente especificadas em sua ISA;
- Entrada, Saída de dados e qualquer outro tipo de serviço que um sistema computacional necessitar é tratada pelo sistema operacional;
- O sistema operacional se comunica com o processador via chamadas do sistema (Syscalls);
- Um código de serviço é especificado e um sinal é enviado ao processador;
- O processador para o que está fazendo no momento e “atende” a chamada do sistema;
- Associado a cada código de serviço há uma rotina que itera com o dispositivo de entrada e saída via interrupções e espaço de memória compartilhada;

4

## Principais Serviços

Serviço	Código (v0)	Argumentos	Resultado
Imprime inteiro	1	\$a0 ← inteiro a imprimir	
Imprime float	2	\$f12 ← float a imprimir	
Imprime double	3	\$f12 ← double a imprimir	
Imprime string	4	\$a0 ← end. da string	
Lê inteiro	5		\$v0 inteiro lido
Lê float	6		\$f0 float lido
Lê double	7		\$f0-\$f1 double lido
Le string	8	\$a0 ← end do buffer \$a1 ← número de caracteres	
Aloca memória	9	\$a0 ← # de bytes a serem alocados	\$v0 end. da memória alocada
Término	10		
Imprime caractere	11	\$a0 ← caractere a imprimir	
Lê caractere	12		\$v0 caractere lido

5

## Principais Serviços

Serviço	Código (v0)	Argumentos	Resultado
Abre arquivo	13	\$a0 ← nome do arq. \$a1 ← flags \$a2 ← modo	\$v0 ← descritor do arquivo
Lê do arquivo	14	\$a0 ← descritor do arq. \$a1 ← end. do buffer \$a2 ← max # de chars lidos	\$v0 ← # de caracteres lidos Negativo se erro
Escreve no arquivo	15	\$a0 ← descritor do arq. \$a1 ← end. do buffer \$a2 ← # de caracteres a escrever	\$v0 ← # de caracteres escritos Negativo se erro
Fecha arquivo	16	\$a0 ← descritor do arq.	
Término com status	17	\$a0 ← código de término	
tempo	30		\$a0 32 bits menos sig., \$a1 32 mais sig.
dorme	32	\$a0 ← tempo em milisecs.	
Imprime inteiro em hexa	34	\$a0 ← inteiro a imprimir	
Imprime inteiro em binário	35	\$a0 ← inteiro a imprimir	

6

## Saída de Dados

```

.data
inteiro: .word 42
real: .float 3.1415
duplo: .double 2.7182818284590452353602874713526
enter: .ascii "\n"
hwerld: .ascii "oi Mundo!"

.macro done
li $v0, 10
syscall
.end_macro

.macro printNewLine
li $v0, 4
la $a0, enter
syscall
syscall
.end_macro
    
```

## Saída de Dados

```

.text
la $s0, inteiro
lw $w0, 0($s0)

la $s1, real
lwc1 $f20, 0($s1)

la $s2, duplo
lwc1 $f22, 0($s2)
lwc1 $f23, 4($s2)

la $s3, hwerld

#imprime inteiro
li $v0, 1 #código de syscall
add $a0, $zero, $s0
syscall
printNewLine

#imprime inteiro em hexadecimal
li $v0, 34 #código de syscall
add $a0, $zero, $s0
syscall
printNewLine

#imprime inteiro em binário
li $v0, 35 #código de syscall
add $a0, $zero, $s0
syscall
printNewLine

#imprime um float
li $v0, 2
mov.s $f10, $f20
syscall
printNewLine

#imprime um double
li $v0, 3
mov.d $f10, $f22
syscall
printNewLine

#imprime string
li $v0, 4
add $a0, $zero, $s3
syscall
printNewLine
    
```

## Entrada de Dados

```

#le inteiros
li $v0, 5
syscall

#le real
li $v0, 6
syscall

#le duplo
li $v0, 7
syscall

#le string
li $v0, 8
addi $a1, $zero, 30
la $a0, $buffer
syscall
    
```

## Bibliografia Comentada



- **PATTERSON, D. A. e HENNESSY, J. L. 2014.** *Organização e Projeto de Computadores – A Interface Hardware/Software*. Elsevier/ Campus 4ª edição.



- **HENNESSY, J. L. e PATTERSON, D. A. 2012.** *Arquitetura de Computadores – Uma Abordagem Quantitativa*. Elsevier/ Campus 5ª edição.

## Bibliografia Comentada



- **MONTEIRO, M. A. 2001.** *Introdução à Organização de Computadores*. s.l.: LTC, 2001.



- **MURDOCCA, M. J. e HEURING, V. P. 2000.** *Introdução à Organização de Computadores*. 2000. 85-352-0684-1.

## Bibliografia Comentada



- **STALLINGS, W. 2002.** *Arquitetura e Organização de Computadores*. 2002.



- **TANENBAUM, A. S. 2007.** *Organização Estruturada de Computadores*. 2007.