

# 2023/1 CCT ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES

[Painel](#) / Meus cursos / [Departamento de Ciência da Computação](#) / [Bacharelado em Ciência da Computação](#) / [2023/1 CC](#)  
/ [2023/1 CCT CCI192-04U AOC0004](#)

## Informações

[Avisos](#)

### Professor

Dr. Yuri Kaszubowski Lopes - Email: yuri.lopes@udesc.br

### Avaliação

$$MS = 0,2*A + 0,3*P_1 + 0,3*P_2 + 0,2*PR$$

Onde:

MS: É a média semestral

A: Média ponderada das atividades solicitadas ao longo da disciplina (Laboratório de Avaliação, Questionários, Tarefas, Laboratórios de Programação, ...).

P<sub>x</sub>: Provas.

PR: Projeto

### Leitura Recomendada

D. Patterson; J. Hennessy. **Organização e Projeto de Computadores: Interface Hardware/Software.** 5<sup>a</sup> Edição. Elsevier Brasil, 2017.



### Aulas

2<sup>a</sup>s-feiras 10:10 - 11:50 sala F302

4<sup>a</sup>s-feiras 10:10 - 11:50 sala F302

Alguns laboratórios práticos no final do semestre poderão ser feitos na F204

### Datas Importantes

4<sup>a</sup>-feira, 19/04: Prova 01 - Sala K203

4<sup>a</sup>-feira, 21/06: Prova 02 - **Sala F212**

2<sup>a</sup>-feira, 10/07: Exame

[Avisos](#)

## Leituras e Documentos

[Hennessy Capítulo 2](#)[Hennessy Capítulo 4](#)[MIPS32 - Architecture For Programmers Volume I-A Introduction to the MIPS32 Architecture](#)[MIPS32 - Architecture for Programmers Volume II-A The MIPS32 Instruction Set Manual \(MIPS MANUAL\)](#)

✓ Concluído

## Semana 01

 [Handout-5.1: Conjuntos de Instruções: MIPS, Registradores, Palavras e tipos de instruções](#)

✓ Concluído

 [Handout-5.2: Lidando com a Memória e Operações Lógicas](#)

✓ Concluído

## Semana 02

 [Handout-5.3: Simuladores MARS/Spim, E/S no MIPS e Imprimindo Strings](#)

✓ Concluído

### Instruções para os Laboratório de programação Assembly para MIPS

Faça as implementações em Assembly para MIPS. Será utilizado a plataforma VPL que usa o montador **Spim**. **Spim** e **MARS** possuem algumas diferenças, assim caso use MARS em sua máquina, após fazer os testes com MARS, faça testes dentro do VPL (clique em executar) antes de enviar para a avaliação.

Quando estiver satisfeito, **clique em avaliar**, o VPL do Moodle dá o resultado na hora.

Passo a passo:

Dentro da questão vá em **Editar**

Dê um nome com extensão .s

Adicione seu código

Clique em salvar: 

Clique em executar: 

Teste seu programa (você pode clicar em executar várias vezes para realizar vários testes)

Se estiver satisfeito clique em avaliar:  **0/3 -1**



Você tem um limite de tentativas de avaliação para receber a nota máxima (no exemplo da figura acima são 3)

Após este limite ser excedido, há uma penalização para cada tentativa extra (no exemplo da figura acima é 1,0 ponto por tentativa extra)

### Instruções e dicas:

Seu programa **deve** ter a extensão .s (por exemplo, q01.s) para o VPL

A tabela abaixo apresenta alguns **códigos syscall**

Não há *subi* no Spim, assim utilize addi com imediato negativo para subtração

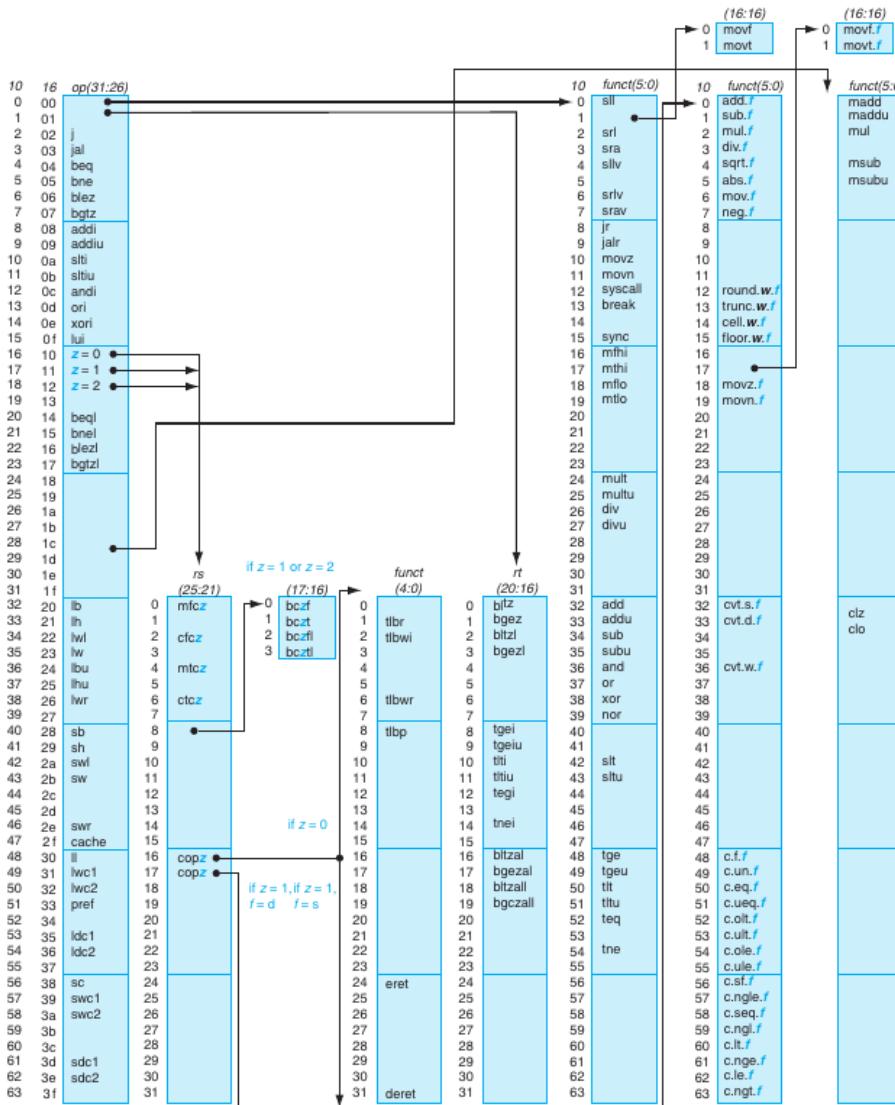
Coloque em .data uma string com \n, assim você pode facilmente adicionar pulos de linha

Códigos syscall:

Descrição	Cód em \$v0	Argumentos	Resultado
Imprimir inteiro	1	\$a0 = inteiro para ser impresso	
Imprimir float	2	\$f12 = float para ser impresso	
Imprimir double	3	\$f12 = double para ser impresso	
Imprimir string	4	\$a0 = endereço de uma string a ser impressa	
Ler inteiro	5		\$v0 contém o inteiro lido
Ler float	6		\$f0 contém o float lido
Ler double	7		\$f0 contém o double lido
Ler string	8	\$a0 = endereço do buffer de entrada \$a1 = número máximo de bytes a ser lido	*Segue o padrão do fgets do C.
Alocar memória heap	9	\$a0 = número de bytes para alocar	\$v0 contém o endereço da memória alocada
Fim (término da execução)	10		
Imprimir char	11	\$a0 = caractere para ser impresso	

Ler char	12	\$f0 contém o char lido
✓ Concluído		

## Mapa dos opcodes do MIPS



 Concluído

Lab 01

Resolva as questões abaixo, todas elas são parte da atividade Lab 01.

 Concluído

VPL Q01 - Programação MIPS - fatorial (5.3 - ex 1)

 Concluído

VPL Q02 - Programação MIPS - soma e média (5.3 - ex 2).

✓ Concluído

VPL Q03 - Programação MIPS - Fibonacci (5.3 - ex 4)

 Concluído

## Semana 03

 [Handout-5.4: Conjuntos de Instruções: Contador de Programa, Branches e Jumps](#)

✓ Concluído

 [Handout-5.5: Chamadas de Função](#)

✓ Concluído

## Semana 04

**Prática em laboratório: Exercícios da aula 5.5**

✓ Concluído

## Semana 05

 [Handout-5.6: Funções não folha e recursão](#)

✓ Concluído

**Lab 02**

Resolva as questões abaixo, todas elas são parte da atividade Lab 02. Todas devem ser implementadas com **funções recursivas**. Entradas e saídas devem ser tratadas na função `main`. Argumentos e retornos passados/recebidos da função devem usar os registradores apropriados (`$a_`, `$v_` e pilha).

✓ Concluído

 [Q01 - Programação MIPS - contar dígitos recursivo](#)

✓ Concluído

 [Q02 - Programação MIPS - somar dígitos recursivo](#)

✓ Concluído

 [Q03 - Programação MIPS - Fibonacci recursivo](#)

✓ Concluído

## Semana 06

 [Handout-6.1: Construindo a CPU: Parte 1](#)

✓ Concluído

 [Handout-6.2: Construindo a CPU: Parte 2](#)

✓ Concluído

## Semanas 07 e 08

 [Handout-6.3: Sinais de Controle](#)

✓ Concluído

 [Revisão p/ P1](#)

✓ Concluído

## Semana 9

 [Handout-6.4: Pipelining](#)

✓ Concluído

 [Handout-6.5: Hazards](#)

✓ Concluído

 [Atividade 04 \(6.4, 6.5 e 6.6\)](#)

✓ Concluído

## Semana 10

 [Handout-6.6: Caminho de Dados com Pipeline](#)

✓ Concluído

## Especificação do Projeto

 [2022-2-Projeto AOC](#)

✓ Concluído

 [PIC10F20x-datasheet](#)

✓ Concluído



 [main](#)

✓ Concluído

 [Escolha de Grupo para o projeto](#)

✓ Concluído

Entrega em pdf pelo Moodle até 26/06/2023.

✓ Concluído

## Semana 11

 [Handout-6.7: Hazards de Dados: Construindo Forwardings \(bypasses\)](#)

✓ Concluído

 [Handout-6.8: Hazards de Controle](#)

✓ Concluído

 [Atividade 05 \(6.7-6.12\)](#)

✓ Concluído

## Semana 12

 [Handout-6.9: Exceções, Interrupções e I/O](#) Concluído [Handout-6.10: Arquiteturas e Abstrações](#) Concluído

## Semana 13

 [Handout-6.11: Paralelismo: Conceitos básicos](#) Concluído [Handout-6.12: GPU: Conceitos Básicos](#) Concluído

## Recuperação da Prova 1

A recuperação (RP1) da Prova 1 (P1), será um questionário no Moodle que deverá ser respondido em sala, na aula de 2ª-feira, 29/05/2023.

Você terá 45 minutos para responder o questionário e haverá dois horários (a escolher, vagas limitadas):

Horário 1: 10:15 às 11:00

Horário 2: 11:05 às 11:50

Use a Atividade: [Escolher Horário para a Recuperação da P1](#). Quem fizer no Horário 1, deverá permanecer em sala até 11:00. Quem fizer no Horário 2 já deverá estar aguardando para entrar na sala às 10:55.

A nova nota da Prova 1 (P1') será:

$$P1' = \max(P1, (P1+RP1)/2)$$

 [Escolher Horário para a Recuperação da P1](#) Concluído [Recuperação P1 - Horário 01](#) Concluído [Recuperação P1 - Horário 02](#)

[Restrito] Disponível se: Você faz parte de **Recuperação P1 - Horário 2**

## Semana 14

 [Handout-7.1: Hierarquia de Memórias e Cache](#) Concluído [Atividade 06 \(7.1-7.4\)](#) Concluído

## Semana 15

 [Handout-7.2: Blocos da Cache e Associatividade](#)

✓ Concluído

 [Handout-7.3: LRU, Caches multinível e Coerência de Cache](#)

✓ Concluído

## Semana 16

 [Handout-7.4: Construção de Memórias](#)

✓ Concluído

 [Handout-8.0: Microcontroladores](#)

✓ Concluído

 [Revisão](#)

✓ Concluído

## Semanas 17, 18 e 19

 [Entrega do Projeto](#)

✓ Concluído

Você acessou como KAUAN HENRIQUE WERLICH (Sair)

Redefinir a demonstração nessa página

Página inicial



Português - Brasil (pt\_br)

Deutsch (de)

English (en)

Español - Internacional (es)

Français (fr)

Italiano (it)

Português - Brasil (pt\_br)

日本語 (ja)

Resumo de retenção de dados

Obter o aplicativo para dispositivos móveis