

Nome: _____

Prova Objetiva 0 (Obj0)**Informações**

- Quantidade de questões: **4**
- Total de pontos: **100**
- Peso no bimestre: **10%**

Instruções

1. Antes de iniciar, confira se esta cópia contém todas as questões e se estão legíveis.
2. Prova individual e com consulta.
3. A correta interpretação das questões faz parte da avaliação.
4. Utilize as próprias folhas da prova como rascunho e espaço para desenvolvimento das soluções.
5. Somente serão avaliadas as respostas na grade de respostas.

Boa Prova!**Tabela de Pontuações – Uso exclusivo do professor!**

Questões	1	2	3	4	Total
Valor	20	30	20	30	100
Pontuação					

Questões

1. [20 pontos] Um usuário está utilizando seu sistema UNIX através de uma sessão de terminal *bash*. Neste momento, ele se encontra no diretório de trabalho `/home/silva/`, que possui os seguintes arquivos: `prova.txt`, `proj`, `cert`, `docs`, `libs` e `bin`. O arquivo chamado `prova.txt` contém o seguinte texto:

```
1 prova de sistemas operacionais 1
2 Prova de Sistemas Operacionais 2
3 PROVA DE SISTEMAS OPERACIONAIS 3
```

Para cada comando entrado abaixo, marque a opção que transcreve a saída do terminal:

(a) `$ pwd`

- A. `C:\Usuários\Ricardo`
- B. `/home/silva/`
- C. `/home/ricardo/sandbox`
- D. `/`
- E. Nenhuma saída

(b) `$ mkdir sandbox && cp prova.txt sandbox/. && ls`

Dica: O operador `&&` faz com que um comando seja executado logo após o fim do outro.

- A. `prova.txt proj cert docs libs bin sandbox`
- B. `proj cert docs libs bin sandbox`
- C. `sandbox/. prova.txt`
- D. `prova.txt --> sandbox/.`
- E. Nenhuma saída.

(c) `$ cd sandbox && echo welcome > ola && ls`

- A. `prova.txt welcome`
- B. `sandbox prova.txt welcome`
- C. `prova.txt ola`
- D. `ola`
- E. `welcome`

(d) `$ cd .. && rm prova.txt && ls`

- A. `prova.txt proj cert docs libs bin sandbox`
- B. `proj cert docs libs bin sandbox`
- C. `prova.txt`
- D. Diretório `".."` não encontrado ou inexistente.
- E. Nenhuma saída.

(e) `$ cd ~ && rm -rf * && ls`

- A. `/home/ricardo`
- B. `/`

C. ~ *

D. Usuário não tem permissão para a operação requisitada.

E. Nenhuma saída.

2. [30 pontos] Para cada código na linguagem C a seguir, indique qual a saída no terminal:

(a)

```
void func1() {  
    if(0)  
        printf("Era verdade... \n");  
    else  
        printf("Verificada falha. \n");  
}
```

A. Era verdade...

B. Verificada falha.

C. Falha de segmentação

D. Erro de compilação.

E. Nenhuma saída.

(b)

```
void func2() {  
    int a = 10, b = 45, c, * p1, * p2;  
    p1 = &b;  
    b = a + (*p1);  
    p2 = p1;  
    c = a * (*p1) + (*p2);  
    p2 = &a;  
    p1 = &c;  
    *p2 = a + (*p1);  
    printf("a: %d, b: %d, c: %d \n", a, b, c);  
}
```

A. a: 10, b: 55, c: 605

B. a: 10, b: 45, c: 215

C. a: 615, b: 55, c: 605

D. Falha de segmentação.

E. Erro de compilação.

(c) Este programa inicia em func3:

```
void aux(int * vals, int len) {  
    while (len > 0) {  
        len--;  
        vals[len-1] += *(vals + len)  
    }  
}  
  
void func3() {  
    int n[] = { 1, 2, 3, 4, 5 };  
    aux(n, 3);  
    print("%d", n[0])  
}
```

- A. 1
- B. 15
- C. 10
- D. 6
- E. Falha de segmentação.

3. [20 pontos] Considere os trechos de um mesmo programa, escrito em linguagem *Assembly* MIPS:

```

53 ERRO_NAO_ENCONTRADO_STR:
54     .asciiz "O arquivo indicado nao foi encontrado."

788     li $v0, 4
789     la $a0, ERRO_NAO_ENCONTRADO_STR
790     syscall

```

A funcionalidade do segundo trecho do programa é pedir para o sistema escrever na tela a frase definida pelo primeiro trecho.

Sabendo disso, leia as seguintes assertivas, e escolha a opção que apresente as verdadeiras:

- (a)
 - I. A instrução `syscall` causa uma interrupção e muda o valor do registrador `$pc`.
 - II. O serviço desejado é indicado pelo valor contido em `$a0`, e o parâmetro do serviço por `$v0`.
 - III. O segundo trecho realiza uma operação do tipo *chamada ao sistema*.
- A. Nenhuma. B. I e II. C. I e III. D. II e III. E. Todas.
- (b)
 - I. Após o `syscall`, o sistema tem a obrigação de escrever a frase na tela.
 - II. Este programa funciona em computadores de quaisquer arquiteturas e sistemas operacionais.
 - III. Este programa funciona em qualquer sistema operacional sobre a arquitetura MIPS.
- A. Nenhuma. B. I e II. C. I e III. D. II e III. E. Todas.

4. [30 pontos] Para cada descrição a seguir, marque a função da biblioteca padrão da Linguagem C que mais está relacionada:

- (a) Escreve uma string formatada na saída padrão do programa (terminal):
A. `printf` B. `puts` C. `scanf` D. `gets` E. `strcat`
- (b) Lê uma string não formatada da entrada padrão do programa (teclado):
A. `printf` B. `puts` C. `scanf` D. `gets` E. `strcat`
- (c) Requisita, ao sistema operacional, espaço de memória dinâmico da heap:
A. `free` B. `malloc` C. `strlen` D. `rand` E. `srand`
- (d) Lê uma string formatada de um arquivo aberto:
A. `strcmp` B. `sscanf` C. `fscanf` D. `sprintf` E. `fprintf`
- (e) Escreve um único caractere em um arquivo aberto:
A. `fgetc` B. `fgets` C. `fputc` D. `fputs` E. `getchar`
- (f) Escreve uma string formatada em outra string:
A. `scanf` B. `fscanf` C. `fprintf` D. `sscanf` E. `sprintf`