

MONITORIA DE INFRAESTRUTURA DE SOFTWARE

Paulo Salgado Filho

Penélope Maria

AULA 1: LINUX + C/C++

▶ Roteiro

▶ Parte 1: Linux

- ▶ História
- ▶ Terminal

▶ Parte 2: c/c++

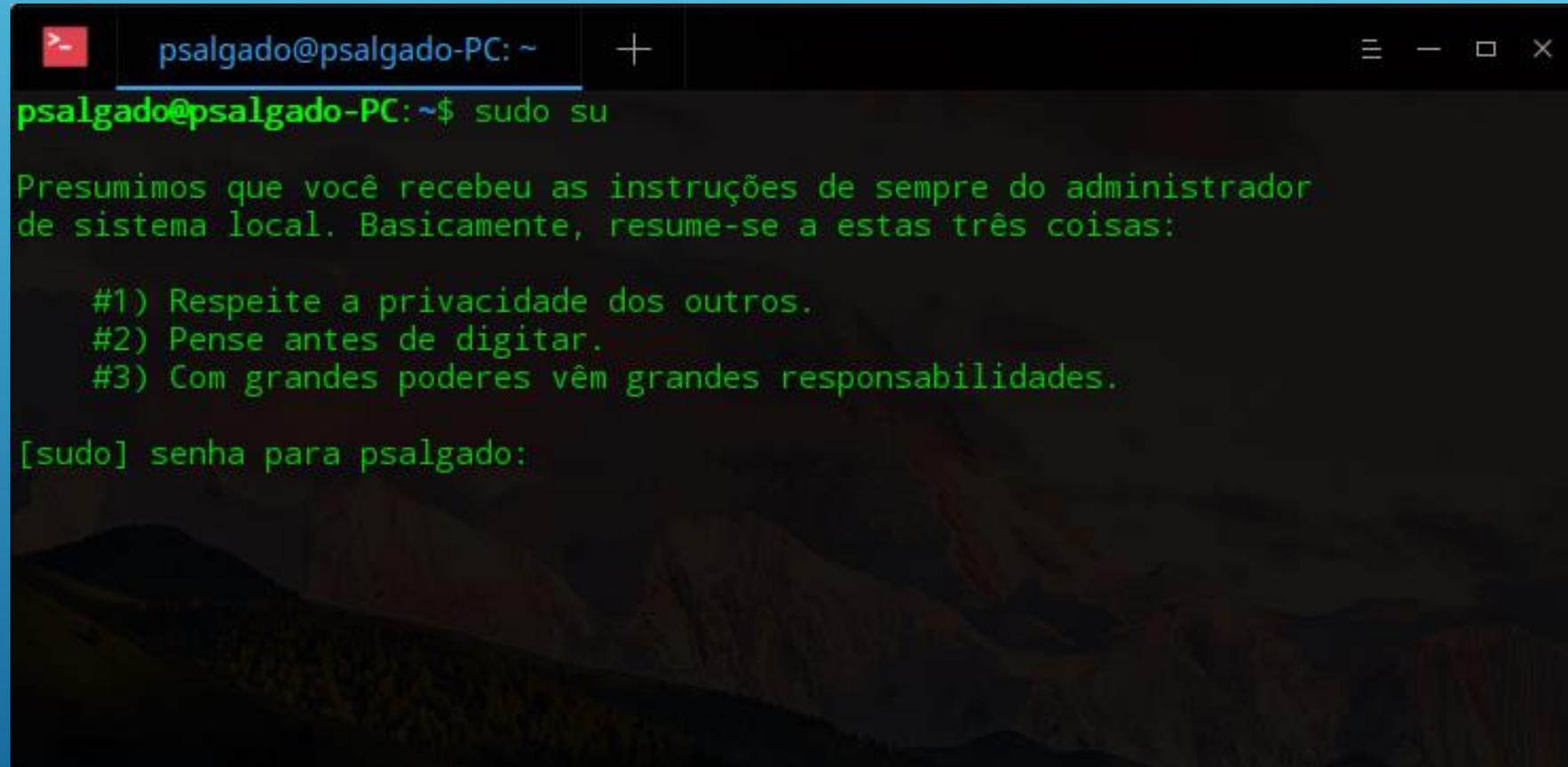
- ▶ História
 - ▶ Estrutura
 - ▶ Compiladores
 - ▶ Ponteiros
- 
- A decorative graphic consisting of several parallel white lines of varying lengths and orientations, located in the bottom right corner of the slide.

A HISTÓRIA

- ▶ Criado pelo Finlandês Linus Torvalds entre 1990 e 1991.
- ▶ Suprir necessidades não atendidas pelo Minix OS.



○ TERMINAL



```
psalgado@psalgado-PC: ~  
psalgado@psalgado-PC:~$ sudo su  
Presumimos que você recebeu as instruções de sempre do administrador  
de sistema local. Basicamente, resume-se a estas três coisas:  
  
#1) Respeite a privacidade dos outros.  
#2) Pense antes de digitar.  
#3) Com grandes poderes vêm grandes responsabilidades.  
  
[sudo] senha para psalgado:
```

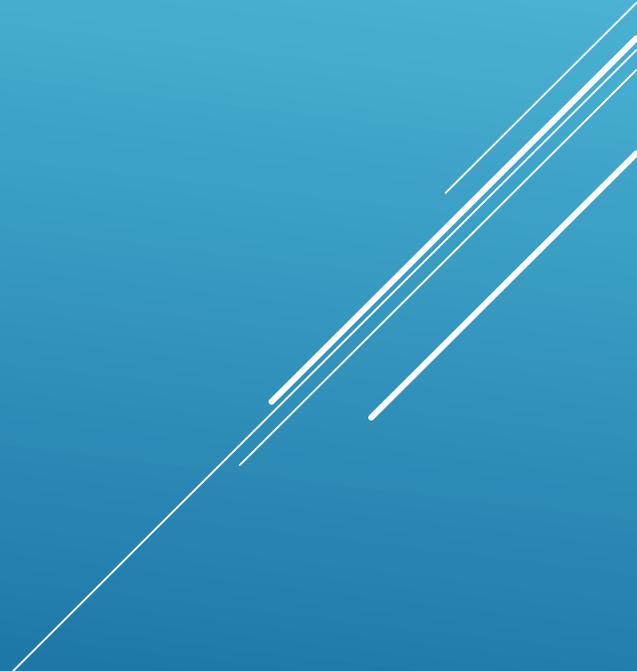
○ TERMINAL

- ▶ Navegação
 - ▶ Criação de diretórios
 - ▶ Mkdir
 - ▶ rmdir
 - ▶ Super user
 - ▶ Sudo
 - ▶ Su
 - ▶ Instalação de programas
 - ▶ Gerenciamento de aplicações
 - ▶ Tópicos de ajuda
 - ▶ Man
 - ▶ --help
- 

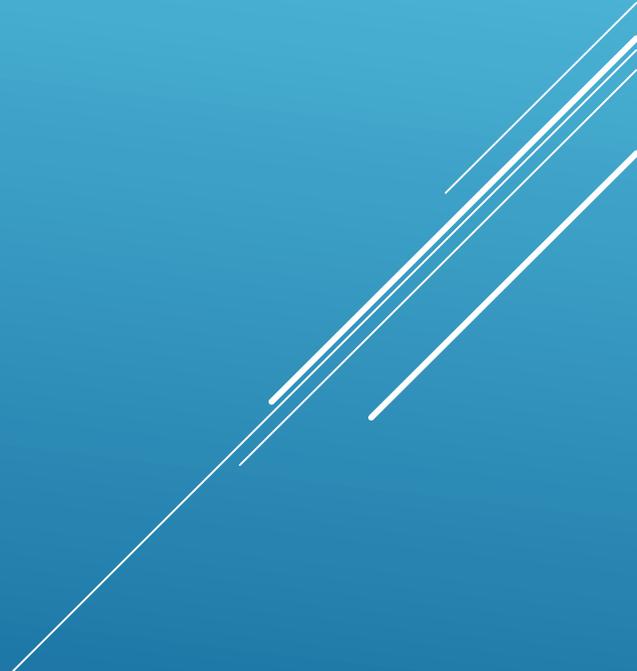
PRATICA

- ▶ Usando o terminal, navegue até sua pasta de documentos e crie a pasta “monitoria”, e use algum editor de texto (Vim, vi, nano, gedit) para criar um documento e nomeie-o como `pratica.c` . Deixe-o em branco.

A LINGUAGEM C

- ▶ Desenvolvida inicialmente por Dennis Ritchie em 1973
 - ▶ Um marco na história da computação
 - ▶ Linguagem imperativa, procedural e estruturada
- 
- A decorative graphic consisting of several parallel white lines of varying lengths, slanted upwards from left to right, located in the bottom right corner of the slide.

A LINGUAGEM C++

- ▶ Criada anos após a linguagem C
 - ▶ “C com orientação a objetos”
- 
- A decorative graphic consisting of several parallel white lines of varying lengths, slanted upwards from left to right, located in the bottom right corner of the slide.

ESTRUTURA DE UM PROGRAMA C

C

```
1  #include <stdio.h>
2
3  int main(){
4      int a =10;
5      char nome[] = "Monitoria de Infra de Software";
6
7      printf("Hello word\n A %s e %d" ,nome,a);
8
9      return 0;
10 }
11
```

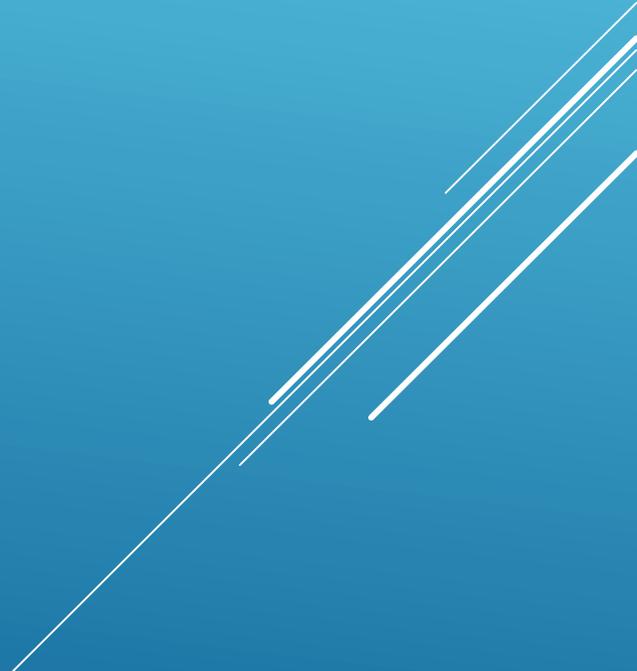
C++

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main(){
6      int a =10;
7      string nome = "Monitoria de Infra de Software";
8
9      cout<<"Hello word"<<endl;
10     cout<<"A "<<nome<<" e "<< a<<endl;
11
12     return 0;
13 }
```

TIPOS DE DADOS

- ▶ Os tipos mais comuns:
 - ▶ Int: `int a = 2;`
 - ▶ Float: `float b = 3,14;`
 - ▶ Char: `char c = 'c'`
- ▶ Para cada tipo podemos especificar um vetor, exemplo:
 - ▶ Int `a[4] = [1,2,3,5]`
 - ▶ Char `b[] = "Linux\n"`

REPETIÇÃO

- ▶ For loop:
 - ▶ For (i=n; condição; incremento)
 - ▶ While
 - ▶ Do{...}while()
- 
- A decorative graphic consisting of several parallel white lines of varying lengths, slanted upwards from left to right, located in the bottom right corner of the slide.

COMPILADORES

- ▶ Compilador C : gcc
 - ▶ Gcc "entrada.c" -o "Saida.c"
- ▶ Compilador C++: g++
 - ▶ G++ "entrada.cpp" -o "saída.cpp"

PRATICA

- ▶ Crie um programa que leia seu nome e idade e imprima seu nome no terminal o numero de vezes correspondente à sua idade
- ▶ Achou fácil? Tente imprimir agora de forma que a string contendo seu nome seja impressa dessa forma:

P

Pa

Paul

Paulo

FUNÇÕES

- ▶ Funcionam exatamente como os métodos em java, com a exceção de que elas não pertencem a uma classe.

```
int or(int a, int b){  
    if(a || b)  
        return 1;  
    else  
        return 0;  
}
```

PONTEIROS

- ▶ Ponteiros são variáveis que guardam o endereço de outra variável
- ▶ Um ponteiro 'aponta' para uma variável quando possui o seu endereço de memória armazenado
- ▶ Exemplo
 - ▶ `Int b = 3;`
 - ▶ `Int *a = &b;`
- ▶ A variável 'a' agora aponta para a variável b

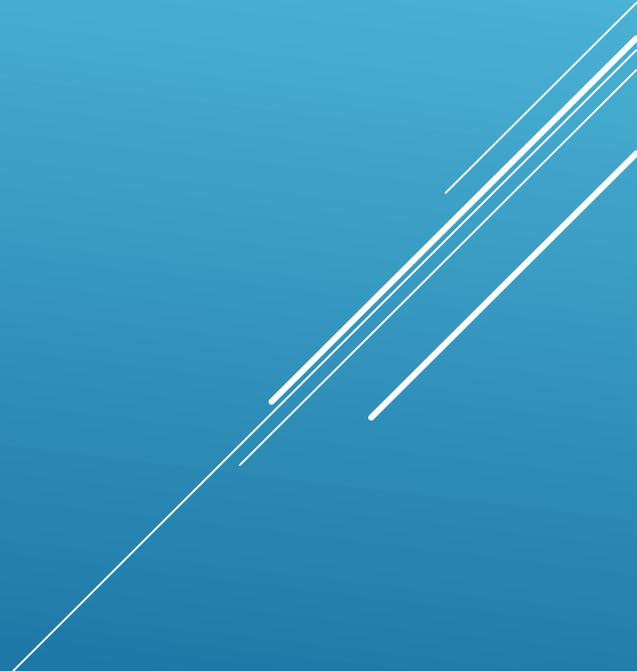
PONTEIROS PRA QUÊ?

- ▶ O conceito de Ponteiros e manipulação de memória, apesar de parecer primitivo e pouco sofisticado, é uma ferramenta poderosa e que será usada com frequência ao trabalharmos com threads.
- 

PRÁTICA

- ▶ Adapte o programa anterior para que ele faça uma chamada de função para imprimir o seu nome N vezes, onde N é sua idade.
 - ▶ Achou fácil? Agora passe apenas um ponteiro para seu nome e para sua idade.
- 

APRENDA MAIS

- ▶ Ganhe prática com terminal, e não o tema, ele é uma ferramenta que potencializa sua produtividade apesar do visual “pouco amigável”
 - ▶ A comunidade Linux têm acentuado seu crescimento nos últimos anos, e o acesso ao conhecimento é cada vez maior
- 

DÚVIDAS ?

