

Argumentos de Linha de Comando

Programação de Computadores 1



Prof. Daniel Saad Nogueira
Nunes

IFB – Instituto Federal de Brasília,
Campus Taguatinga



Sumário

- 1 Introdução
- 2 Argumentos de linha de comando



Sumário

1 Introdução



Introdução

- Alguns programas admitem *argumentos* que podem ser passados por linha de comando.
- Por exemplo, o comando `ls` lista os arquivos do diretório atual.
- Se o comando é invocado com o argumento `-l`, como em `ls -l`, além de listar os arquivos, também são impressas outras informações, como:
 - ▶ Permissões.
 - ▶ Dono do arquivo.
 - ▶ Tamanho em bytes.
 - ▶ Data de última modificação.
 - ▶ ...



Introdução

- Argumentos passados por linha de comando podem ser úteis para modificar o comportamento padrão de um programa ou indicar outro tipo de informação, como por exemplo um caminho de arquivo que deva ser tratado pelo programa.
- Exemplo: `cat ../arquivo.txt` imprime na tela o conteúdo do arquivo `arquivo.txt` que está um diretório acima do atual.



Introdução

- A linguagem C possibilita que argumentos sejam capturados da linha de comando e posteriormente tratados pelo programa.



Sumário

2 Argumentos de linha de comando



Argumentos de linha de comando

- Para suportar a captura de argumentos passados via linha de comando, só precisamos alterar os parâmetros da função `main` da seguinte forma:

```
1  int main(int argc, char *argv[]) {  
2      return 0;  
3  }
```

- `argc` : indica a quantidade de argumentos passados (contador de argumentos).
- `argv` : vetor de argumentos (strings) propriamente dito, tem tamanho `argc` .



Argumentos de linha de comando

- O programa a seguir imprime todos os argumentos passados via linha de comando para ele:

```
1  #include <stdio.h>
2
3  int main(int argc, char *argv[]) {
4      for (int i = 0; i < argc; i++) {
5          printf("argv[%d] = %s\n", i, argv[i]);
6      }
7      return 0;
8  }
```



Argumentos de linha de comando

- É importante notar que `argv[0]` sempre corresponde ao nome do executável.
- Assim, caso o programa seja invocado como `./programa arg1 arg2 arg3`, será impresso:

```
1 ./programa
2 arg1
3 arg2
4 arg3
```



Argumentos de linha de comando

- Cabe o programador receber as strings e tratá-las para dar o comportamento desejado ao programa.