

# Programação Estruturada

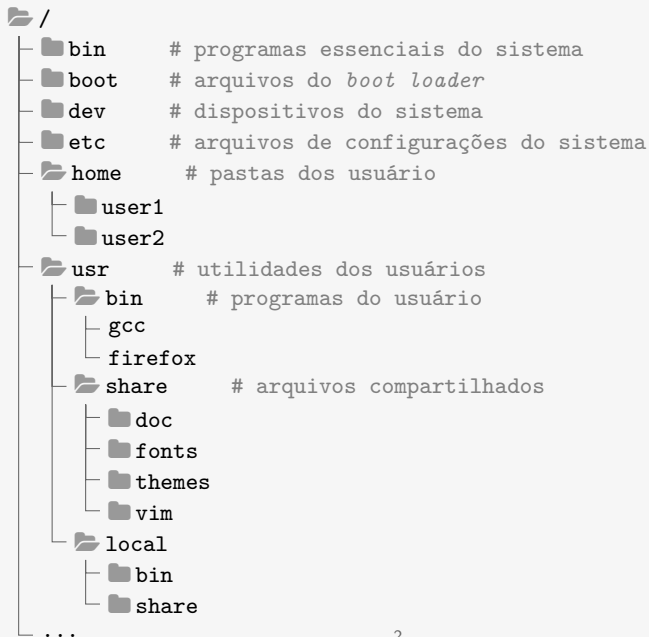
## Laboratório - 01

Maycon Sambinelli

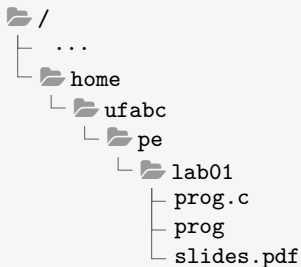
Universidade Federal do ABC

Atualizado em: 2024-10-07 18:32

# Hierarquia do sistema de arquivos



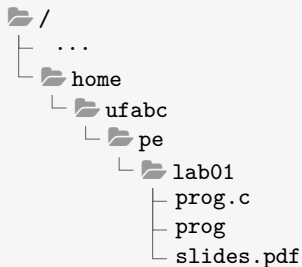
# Diretórios



Cada diretório (e arquivo) pode ser endereçado listando o caminho de diretórios da raiz até o diretório em questão.

- Os diretórios desse caminho são separados por /
- Espaços em nomes de diretório (e arquivos) devem ser escapados com \
- Chamamos esse endereçamento de **endereçamento absoluto**

# Diretórios



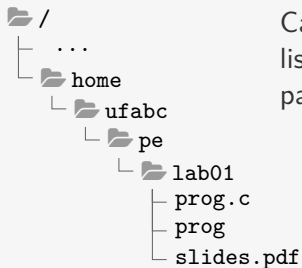
Cada diretório (e arquivo) pode ser endereçado listando o caminho de diretórios da raiz até o diretório em questão.

- Os diretórios desse caminho são separados por **/**
- Espaços em nomes de diretório (e arquivos) devem ser escapados com **\**
- Chamamos esse endereçamento de **endereçamento absoluto**

## EXEMPLOS:

- **/home/ufabc/pe/lab01/**
- **/home/ufabc/pe/lab01/slides.pdf**
- **/home/ufabc/pe/lab01/prog**

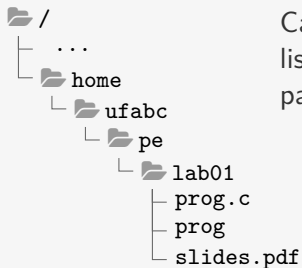
# Diretórios



Cada diretório (e arquivo) pode ser endereçado listando o caminho de diretórios relativo a pasta atual

- Os diretórios atual é denotado por `.`
- O diretório superior é denotado por `..`
- Chamamos esse endereçamento de **endereçamento relativo**

# Diretórios



Cada diretório (e arquivo) pode ser endereçado listando o caminho de diretórios relativo a pasta atual

- Os diretórios atual é denotado por `.`
- O diretório superior é denotado por `..`
- Chamamos esse endereçamento de **endereçamento relativo**

**EXEMPLOS (ASSUMINDO QUE ESTAMOS NA PASTA `pe`):**

- `./lab01/prog.c`
- `../../home`

## Outras considerações

- É comum chamarmos de **path** o endereço (relativo ou absoluto) para um arquivo ou diretório
- Podemos usar `~` como um atalho para o endereço do diretório *home* do usuário

# Comandos Básicos

- `pwd` – Imprime o endereço absoluto do diretório atual



# Comandos Básicos

- `pwd` – Imprime o endereço absoluto do diretório atual
- `cd path` (Change Directory) – muda o diretório de trabalho para o diretório de endereço `path`

# Comandos Básicos

- `pwd` – Imprime o endereço absoluto do diretório atual
- `cd path` (Change Directory) – muda o diretório de trabalho para o diretório de endereço `path`
- `ls [OPTION] [path]` (List) – lista o conteúdo de um diretório

# Comandos Básicos

- `pwd` – Imprime o endereço absoluto do diretório atual
- `cd path` (Change Directory) – muda o diretório de trabalho para o diretório de endereço `path`
- `ls [OPTION] [path]` (List) – lista o conteúdo de um diretório
  - Se um `path` for fornecido, então lista o conteúdo do diretório `path`. Caso contrário, lista o conteúdo do diretório de trabalho atual.

# Comandos Básicos

- `pwd` – Imprime o endereço absoluto do diretório atual
- `cd path` (Change Directory) – muda o diretório de trabalho para o diretório de endereço `path`
- `ls [OPTION] [path]` (List) – lista o conteúdo de um diretório
  - Se um `path` for fornecido, então lista o conteúdo do diretório `path`. Caso contrário, lista o conteúdo do diretório de trabalho atual.
  - `OPTION` muitas opções podem ser passadas para esse comando para modificar o comportamento do programa. Algumas delas são:

# Comandos Básicos

- `pwd` – Imprime o endereço absoluto do diretório atual
- `cd path` (Change Directory) – muda o diretório de trabalho para o diretório de endereço `path`
- `ls [OPTION] [path]` (List) – lista o conteúdo de um diretório
  - Se um `path` for fornecido, então lista o conteúdo do diretório `path`. Caso contrário, lista o conteúdo do diretório de trabalho atual.
  - `OPTION` muitas opções podem ser passadas para esse comando para modificar o comportamento do programa. Algumas delas são:
    - `-l` – para exibir uma versão detalhada de informações sobre os arquivos

# Comandos Básicos

- `pwd` – Imprime o endereço absoluto do diretório atual
- `cd path` (Change Directory) – muda o diretório de trabalho para o diretório de endereço `path`
- `ls [OPTION] [path]` (List) – lista o conteúdo de um diretório
  - Se um `path` for fornecido, então lista o conteúdo do diretório `path`. Caso contrário, lista o conteúdo do diretório de trabalho atual.
  - `OPTION` muitas opções podem ser passadas para esse comando para modificar o comportamento do programa. Algumas delas são:
    - `-l` – para exibir uma versão detalhada de informações sobre os arquivos
    - `-a` – para exibir inclusive os arquivos ocultos. Em sistemas UNIX, qualquer arquivo cujo nome comece com `.`, como por exemplo, `.my_secretes.txt`, é considerado um arquivo oculto.

# Comandos Básicos

- `pwd` – Imprime o endereço absoluto do diretório atual
- `cd path` (Change Directory) – muda o diretório de trabalho para o diretório de endereço `path`
- `ls [OPTION] [path]` (List) – lista o conteúdo de um diretório
  - Se um `path` for fornecido, então lista o conteúdo do diretório `path`. Caso contrário, lista o conteúdo do diretório de trabalho atual.
  - `OPTION` muitas opções podem ser passadas para esse comando para modificar o comportamento do programa. Algumas delas são:
    - `-l` – para exibir uma versão detalhada de informações sobre os arquivos
    - `-a` – para exibir inclusive os arquivos ocultos. Em sistemas UNIX, qualquer arquivo cujo nome comece com `.`, como por exemplo, `.my_secretes.txt`, é considerado um arquivo oculto.
    - `-h` – exibir a página de ajuda programa, que é uma versão da documentação do bem resumida. A maioria dos programas de sistemas UNIX aceitam receber a opção `-h` (ou `--help`) como opção para exibir a página de ajuda.

# Comandos Básicos

- `cat [path]` – imprime o conteúdo do arquivo `path`



# Comandos Básicos

- `cat [path]` – imprime o conteúdo do arquivo `path`
- `rm [OPTION] [path]` – remove o arquivo (ou diretório) de endereço `path`

# Comandos Básicos

- `cat [path]` – imprime o conteúdo do arquivo `path`
- `rm [OPTION] [path]` – remove o arquivo (ou diretório) de endereço `path`
  - Se `path` for o endereço de um diretório, então será necessário passar as opções `-rf`. Exemplo: `rm -rf /Documents/pe`

# Comandos Básicos

- `cat [path]` – imprime o conteúdo do arquivo `path`
- `rm [OPTION] [path]` – remove o arquivo (ou diretório) de endereço `path`
  - Se `path` for o endereço de um diretório, então será necessário passar as opções `-rf`. Exemplo: `rm -rf /Documents/pe`
  - `OPTION` o comando `rm` possui muitas opções. Descreverei apenas os mais comuns:

# Comandos Básicos

- `cat [path]` – imprime o conteúdo do arquivo `path`
- `rm [OPTION] [path]` – remove o arquivo (ou diretório) de endereço `path`
  - Se `path` for o endereço de um diretório, então será necessário passar as opções `-rf`. Exemplo: `rm -rf /Documents/pe`
  - `OPTION` o comando `rm` possui muitas opções. Descreverei apenas os mais comuns:
    - `-r` – remove diretórios e seus conteúdos de forma recursiva

# Comandos Básicos

- `cat [path]` – imprime o conteúdo do arquivo `path`
- `rm [OPTION] [path]` – remove o arquivo (ou diretório) de endereço `path`
  - Se `path` for o endereço de um diretório, então será necessário passar as opções `-rf`. Exemplo: `rm -rf /Documents/pe`
  - `OPTION` o comando `rm` possui muitas opções. Descreverei apenas os mais comuns:
    - `-r` – remove diretórios e seus conteúdos de forma recursiva
    - `-f` – ignore arquivos não existente

# Comandos Básicos

- `cat [path]` – imprime o conteúdo do arquivo `path`
- `rm [OPTION] [path]` – remove o arquivo (ou diretório) de endereço `path`
  - Se `path` for o endereço de um diretório, então será necessário passar as opções `-rf`. Exemplo: `rm -rf /Documents/pe`
  - `OPTION` o comando `rm` possui muitas opções. Descreverei apenas os mais comuns:
    - `-r` – remove diretórios e seus conteúdos de forma recursiva
    - `-f` – ignore arquivos não existente
    - `--help` – mostra a página de ajuda

# Comandos Básicos

- `cat [path]` – imprime o conteúdo do arquivo `path`
- `rm [OPTION] [path]` – remove o arquivo (ou diretório) de endereço `path`
  - Se `path` for o endereço de um diretório, então será necessário passar as opções `-rf`. Exemplo: `rm -rf /Documents/pe`
  - `OPTION` o comando `rm` possui muitas opções. Descreverei apenas os mais comuns:
    - `-r` – remove diretórios e seus conteúdos de forma recursiva
    - `-f` – ignore arquivos não existente
    - `--help` – mostra a página de ajuda
  - **Obs:** Opções de “uma menos”, geralmente, podem ser agrupadas sob um único `-`, como fizemos acima. No exemplo, escrevemos `-rf`, mas poderíamos ter escrito `rm -r -f /Documents/pe`

# Comandos Básicos

- `cp [OPTION] orig1 [orig2] [...] [dest]` – copia arquivos e diretórios



# Comandos Básicos

- `cp [OPTION] orig1 [orig2] [...] [dest]` – copia arquivos e diretórios
  - `orig1 [orig2] [...]` – path dos arquivos a serem copiados. Ao menos um path de origem deve ser fornecido.

# Comandos Básicos

- `cp [OPTION] orig1 [orig2] [...] [dest]` – copia arquivos e diretórios
  - `orig1 [orig2] [...]` – path dos arquivos a serem copiados. Ao menos um path de origem deve ser fornecido.
  - `dest` – path do diretório de destino dos arquivos

# Comandos Básicos

- `cp [OPTION] orig1 [orig2] [...] [dest]` – copia arquivos e diretórios
  - `orig1 [orig2] [...]` – path dos arquivos a serem copiados. Ao menos um path de origem deve ser fornecido.
  - `dest` – path do diretório de destino dos arquivos
  - `OPTION` – opções

# Comandos Básicos

- `cp [OPTION] orig1 [orig2] [...] [dest]` – copia arquivos e diretórios
  - `orig1 [orig2] [...]` – path dos arquivos a serem copiados. Ao menos um path de origem deve ser fornecido.
  - `dest` – path do diretório de destino dos arquivos
  - `OPTION` – opções
    - `-r` – caso um dos paths de origem seja um diretório, é necessário habilitar essa opção para que o conteúdo do diretório seja copiado recursivamente.

# Comandos Básicos

- `cp [OPTION] orig1 [orig2] [...] [dest]` – copia arquivos e diretórios
  - `orig1 [orig2] [...]` – path dos arquivos a serem copiados. Ao menos um path de origem deve ser fornecido.
  - `dest` – path do diretório de destino dos arquivos
  - `OPTION` – opções
    - `-r` – caso um dos paths de origem seja um diretório, é necessário habilitar essa opção para que o conteúdo do diretório seja copiado recursivamente.
    - `-v` – modo verboso: mostra informações sobre o progresso da execução do comando

# Comandos Básicos

- `cp [OPTION] orig1 [orig2] [...] [dest]` – copia arquivos e diretórios
  - `orig1 [orig2] [...]` – path dos arquivos a serem copiados. Ao menos um path de origem deve ser fornecido.
  - `dest` – path do diretório de destino dos arquivos
  - `OPTION` – opções
    - `-r` – caso um dos paths de origem seja um diretório, é necessário habilitar essa opção para que o conteúdo do diretório seja copiado recursivamente.
    - `-v` – modo verboso: mostra informações sobre o progresso da execução do comando
    - `--help` – para exibir a página de ajuda e ver a lista completa de opções.

# Comandos Básicos

- `cp [OPTION] orig1 [orig2] [...] [dest]` – copia arquivos e diretórios
  - `orig1 [orig2] [...]` – path dos arquivos a serem copiados. Ao menos um path de origem deve ser fornecido.
  - `dest` – path do diretório de destino dos arquivos
  - `OPTION` – opções
    - `-r` – caso um dos paths de origem seja um diretório, é necessário habilitar essa opção para que o conteúdo do diretório seja copiado recursivamente.
    - `-v` – modo verboso: mostra informações sobre o progresso da execução do comando
    - `--help` – para exibir a página de ajuda e ver a lista completa de opções.
- `mv [OPTION] orig1 [orig2] [...] [dest]` – move arquivos e diretórios

# Comandos Básicos

- `cp [OPTION] orig1 [orig2] [...] [dest]` – copia arquivos e diretórios
  - `orig1 [orig2] [...]` – path dos arquivos a serem copiados. Ao menos um path de origem deve ser fornecido.
  - `dest` – path do diretório de destino dos arquivos
  - `OPTION` – opções
    - `-r` – caso um dos paths de origem seja um diretório, é necessário habilitar essa opção para que o conteúdo do diretório seja copiado recursivamente.
    - `-v` – modo verboso: mostra informações sobre o progresso da execução do comando
    - `--help` – para exibir a página de ajuda e ver a lista completa de opções.
- `mv [OPTION] orig1 [orig2] [...] [dest]` – move arquivos e diretórios
  - Opções e parâmetros análogos ao comando `cp`, mas move os arquivos ao invés de copiá-los



## Comandos Básicos

- `mkdir [path]` – (make directory) cria um diretório `path`.

# Comandos Básicos

- `mkdir [path]` – (make directory) cria um diretório `path`.
  - **Exemplos:**

# Comandos Básicos

- `mkdir [path]` – (make directory) cria um diretório `path`.
  - **Exemplos:**
    - `mkdir pe` – cria um diretório chamado `pe` no diretório de trabalho atual

# Comandos Básicos

- `mkdir [path]` – (make directory) cria um diretório `path`.
  - **Exemplos:**
    - `mkdir pe` – cria um diretório chamado `pe` no diretório de trabalho atual
    - `mkdir ../lab01` – cria um diretório chamado `lab01` no diretório acima

# Comandos Básicos

- `mkdir [path]` – (make directory) cria um diretório `path`.
  - **Exemplos:**
    - `mkdir pe` – cria um diretório chamado `pe` no diretório de trabalho atual
    - `mkdir ../lab01` – cria um diretório chamado `lab01` no diretório acima
    - `mkdir /home/ufabc/Documents/pe/lab02` – cria um diretório chamado `lab02` dentro da pasta `/home/ufabc/Documents/pe`

# Comandos Básicos

- `mkdir [path]` – (make directory) cria um diretório `path`.
  - **Exemplos:**
    - `mkdir pe` – cria um diretório chamado `pe` no diretório de trabalho atual
    - `mkdir ../lab01` – cria um diretório chamado `lab01` no diretório acima
    - `mkdir /home/ufabc/Documents/pe/lab02` – cria um diretório chamado `lab02` dentro da pasta `/home/ufabc/Documents/pe`
- `touch [path]` – cria um arquivo vazio em `path`

# Comandos Básicos

- `mkdir [path]` – (make directory) cria um diretório `path`.
  - **Exemplos:**
    - `mkdir pe` – cria um diretório chamado `pe` no diretório de trabalho atual
    - `mkdir ../lab01` – cria um diretório chamado `lab01` no diretório acima
    - `mkdir /home/ufabc/Documents/pe/lab02` – cria um diretório chamado `lab02` dentro da pasta `/home/ufabc/Documents/pe`
- `touch [path]` – cria um arquivo vazio em `path`
  - Parâmetros e opções similares à `mkdir`

## Comandos Básicos

- O terminal tem um autocompletar poderoso que completa comandos e paths. Para disparar o autocomplete, basta apertar a tecla `<tab>`.



## Comandos Básicos

- O terminal tem um autocompletar poderoso que completa comandos e paths. Para disparar o autocomplete, basta apertar a tecla `<tab>`.
- **Caracteres curingas** - o terminal possui alguns caracteres curingas. O principal deles é o `*` que substitui um conjunto de zero ou mais caracteres.

# Comandos Básicos

- O terminal tem um autocompletar poderoso que completa comandos e paths. Para disparar o autocomplete, basta apertar a tecla `<tab>`.
- **Caracteres curingas** - o terminal possui alguns caracteres curingas. O principal deles é o `*` que substitui um conjunto de zero ou mais caracteres.
  - **Exemplo:** Imagine que no diretório de trabalho atual, tivéssemos vários arquivos cujos nomes seguem o padrão: `lab01`, `lab02`, `lab03`, .... Podemos escrever `cp -rf lab* ../../backup/labs/` para copiar todos esses arquivos para `../../backup/labs/`

# Compilando no terminal

Compilando (no terminal):

```
gcc -std=c99 -Wall -Wvla -Werror -g -lm programa.c -o programa
```

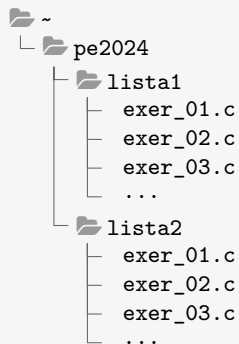
Flags:

- std=c99: usa o padrão C99
- Wall: dá mais *warnings* de compilação
- Wvla: *warnings* para *variable length arrays*
- Werror: *warnings* viram erros de compilação
- g: permite usar gdb e valgrind
- lm: permite usar funções matemáticas
- o: define o nome do programa

Executando o programa:

- `./programa`

# Mãos a Obra



## Menu do dia

- Questão 5
- Questão 1
- Questão 2
- Questão 3