

Arquivos Idênticos

Tarefa

A sua tarefa é determinar se os arquivos passados como argumento para o seu programa são **iguais** ou **diferentes**.

Para determinar se arquivos são iguais ou diferentes você deve comparar todos os bytes dos dois arquivos. E, claro, se os tamanhos forem diferentes você já pode considerar que os arquivos são diferentes.

Para tratar os parâmetros você deve utilizar os parâmetros `argc` e `argv`, da `main`. A maneira mais comum é declarando a sua função principal da seguinte maneira:

```
int main(int argc, char **argv)
{
}
```

Entrada

O seu programa será invocado com parâmetros na linha de comando, que representam arquivos, por exemplo:

```
./solucao arquivo1 arquivo2 arquivo3
```

Você pode considerar que serão passados até 2000 arquivos como argumento para o seu programa.

Cada arquivo pode conter um tamanho que varia de 1 byte até 300 MegaBytes. E cada arquivo pode ser tanto binário como de texto.

Saída

Para cada par você deverá imprimir os nomes dos arquivos e a string **iguais** quando os arquivos forem idênticos e **diferentes** se tiverem ao menos 1 byte diferente.

Exemplos

Exemplo de entrada

```
./solucao /bin/ls /bin/grep /bin/bash
```

Saída para o exemplo

```
/bin/ls /bin/grep diferentes
/bin/ls /bin/bash diferentes
/bin/grep /bin/bash diferentes
```

Exemplo de entrada

```
date > /tmp/hora
cp /tmp/hora /tmp/hora2
date -R > /tmp/hora3
./solucao /tmp/hora /tmp/hora2 /tmp/hora3 /bin/bash
```

Saída para o exemplo

```
/tmp/hora /tmp/hora2 iguais
/tmp/hora /tmp/hora3 diferentes
```

```
/tmp/hora /bin/bash diferentes
/tmp/hora2 /tmp/hora3 diferentes
/tmp/hora2 /bin/bash diferentes
/tmp/hora3 /bin/bash diferentes
```

Exemplo de entrada

```
echo a > /tmp/a
echo b > /tmp/b
echo -n a > /tmp/an
./solucao /tmp/a /tmp/b /tmp/an
```

Saída para o exemplo

```
/tmp/a /tmp/b diferentes
/tmp/a /tmp/an diferentes
/tmp/b /tmp/an diferentes
```

DICA

1. A dica importante para gerar a saída na ordem esperada é fixar um arquivo e percorrer todos os arquivos para frente dele (pois vetor `argv` é indexado de 1 até N , o que simplifica)
2. Você também deve se familiarizar com as funções `ftell(3)`; `fseek(3)`; `fread(3)`
 - Recomenda-se que o vídeo Manipulando Arquivos Binários e de Texto em C - FSO 2020-2

Author: Bruno Ribas