

## MC833 - 2s2017

Bruno Orsi Berton

RA 150573 - Turma B

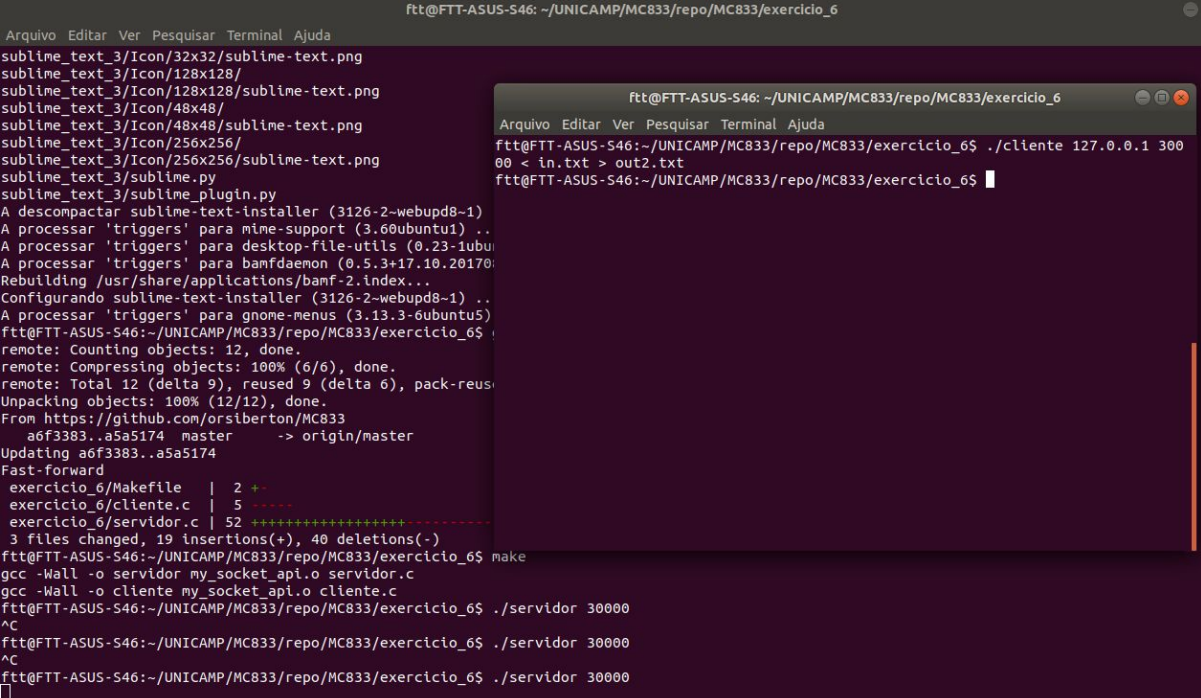
Fábio Takahashi Tanniguchi

RA 145980 - Turma A

1) Modifique o programa cliente de eco da atividade 4 para que este receba como entrada e envie ao servidor não mais linhas digitadas pelo usuário, mas linhas de um arquivo texto qualquer (O arquivo será passado utilizando o caracter de redirecionamento '<'). O cliente continuará recebendo o eco enviado pelo servidor, que deverá ser escrito em um arquivo (O arquivo será criado utilizando o caracter de redirecionamento '>'). Seu programa deverá necessariamente utilizar ou a função select ou a função poll. **\*IMPORTANTE\***: O código será testado com arquivos texto de cerca de 20000 linhas e 1MB. Para certificar que seu código está funcionando corretamente, use arquivos com tamanhos próximos a esses durante os testes.

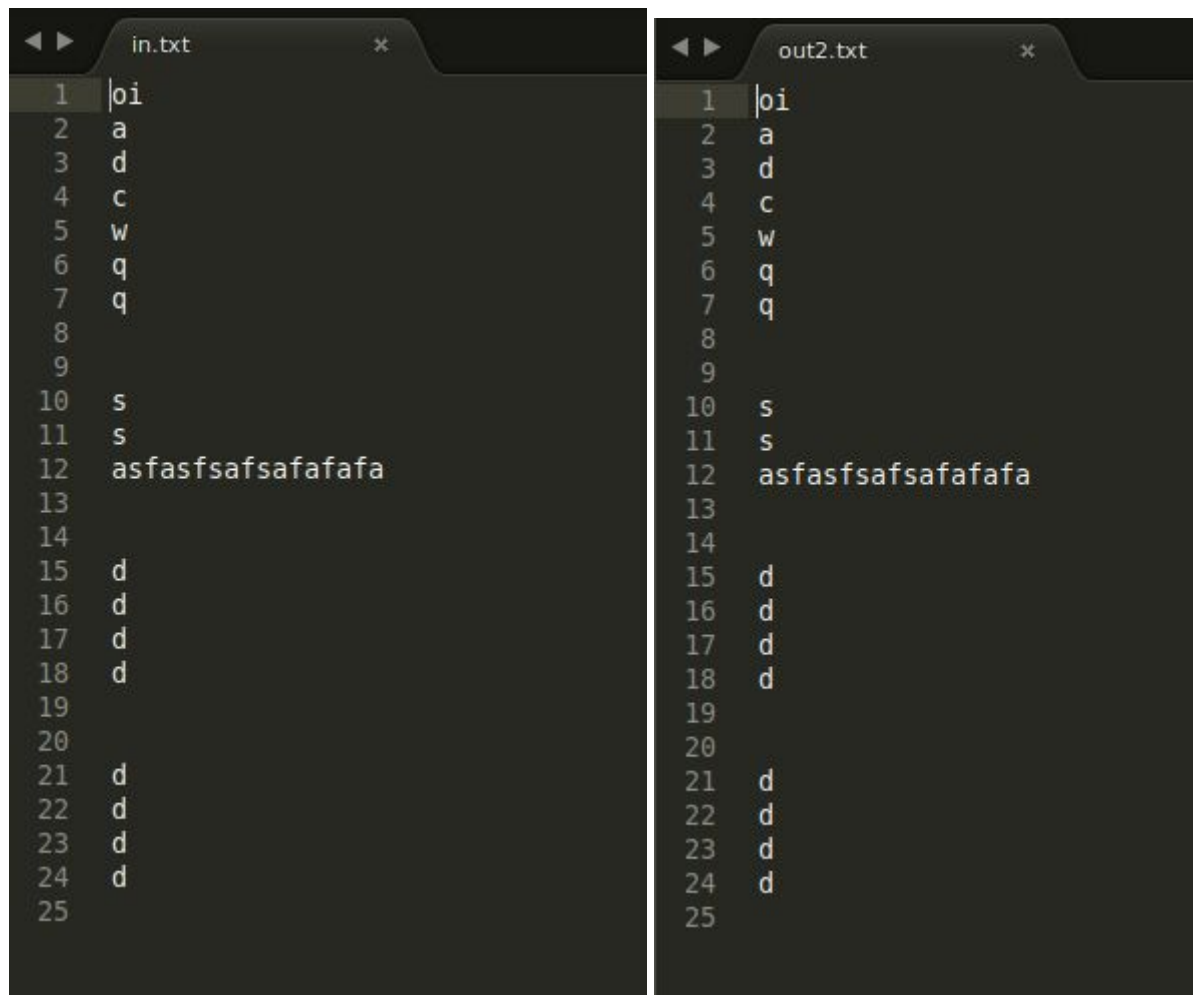
Cada linha deve ser enviada separadamente para o servidor e elas não necessariamente precisam constituir comandos do Unix, porque o servidor não deverá executá-las. O servidor só irá enviá-las de volta para o cliente.

O cliente deve finalizar sua execução assim que tiver recebido todo o arquivo ecoado pelo servidor.



```
ftt@FTT-ASUS-S46: ~/UNICAMP/MC833/repo/MC833/exercicio_6
Arquivo Editar Ver Pesquisar Terminal Ajuda
sublime_text_3/Icon/32x32/sublime-text.png
sublime_text_3/Icon/128x128/
sublime_text_3/Icon/128x128/sublime-text.png
sublime_text_3/Icon/48x48/
sublime_text_3/Icon/48x48/sublime-text.png
sublime_text_3/Icon/256x256/
sublime_text_3/Icon/256x256/sublime-text.png
sublime_text_3/sublime.py
sublime_text_3/sublime_plugin.py
A descompactar sublime-text-installer (3126-2-webupd8-1) ..
A processar 'triggers' para mime-support (3.60ubuntu1) ..
A processar 'triggers' para desktop-file-utils (0.23-1ubu
A processar 'triggers' para bamfdaemon (0.5.3+17.10.20170
Rebuilding /usr/share/applications/bamf-2.index...
Configurando sublime-text-installer (3126-2-webupd8-1) ..
A processar 'triggers' para gnome-menus (3.13.3-6ubuntu5)
ftt@FTT-ASUS-S46:~/UNICAMP/MC833/repo/MC833/exercicio_6$
remote: Counting objects: 12, done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 12 (delta 9), reused 9 (delta 6), pack-reus
Unpacking objects: 100% (12/12), done.
From https://github.com/orsibert/MC833
   a6f3383..a5a5174 master    -> origin/master
Updating a6f3383..a5a5174
Fast-forward
 exercicio_6/Makefile      | 2 +
 exercicio_6/cliente.c     | 5 +
 exercicio_6/servidor.c    | 52 +
 3 files changed, 19 insertions(+), 40 deletions(-)
ftt@FTT-ASUS-S46:~/UNICAMP/MC833/repo/MC833/exercicio_6$ make
gcc -Wall -o servidor my_socket_api.o servidor.c
gcc -Wall -o cliente my_socket_api.o cliente.c
ftt@FTT-ASUS-S46:~/UNICAMP/MC833/repo/MC833/exercicio_6$ ./servidor 30000
^C
ftt@FTT-ASUS-S46:~/UNICAMP/MC833/repo/MC833/exercicio_6$ ./servidor 30000
^C
ftt@FTT-ASUS-S46:~/UNICAMP/MC833/repo/MC833/exercicio_6$ ./servidor 30000
^C
```

Acima, é possível visualizar a execução do cliente (com entrada in.txt e saída out2.txt) e a do servidor.

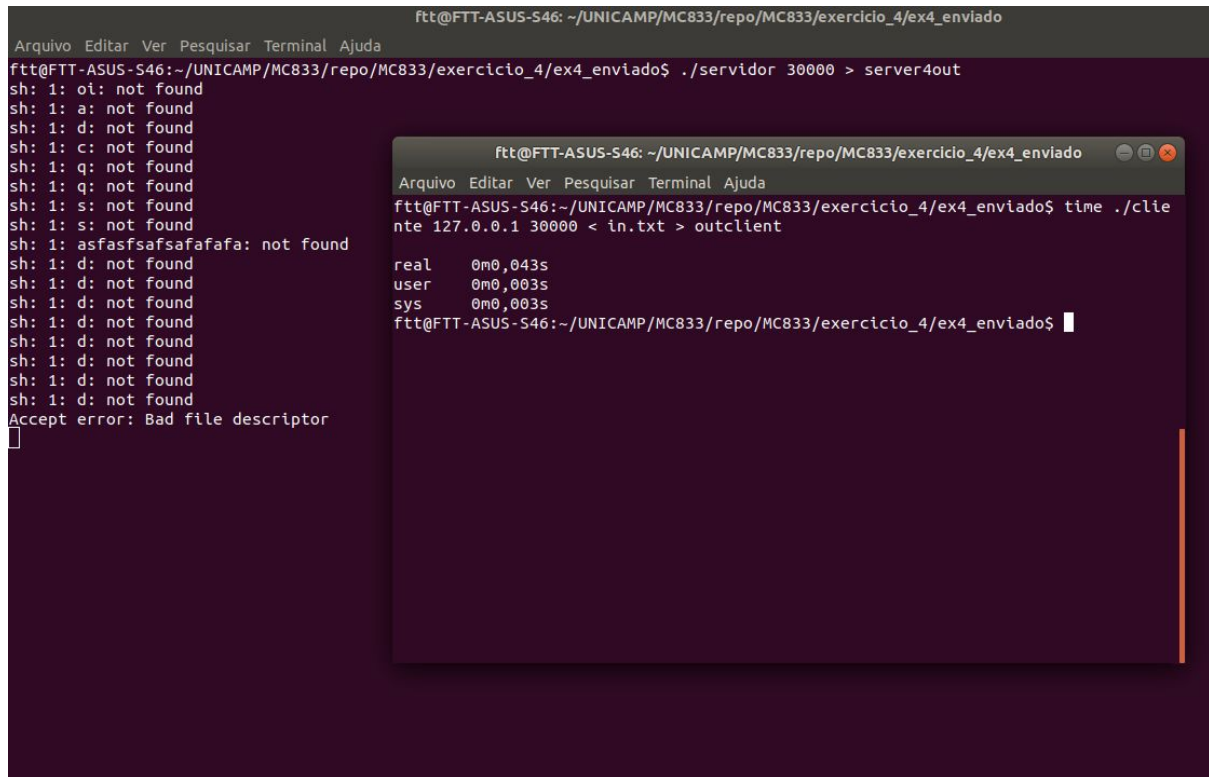


The image displays two side-by-side text editors. The left editor, titled 'in.txt', and the right editor, titled 'out2.txt', both show the same 25-line text. The text is as follows:

```
1 |oi
2 a
3 d
4 c
5 w
6 q
7 q
8
9
10 s
11 s
12 asfasfsafsafafafa
13
14
15 d
16 d
17 d
18 d
19
20
21 d
22 d
23 d
24 d
25
```

Acima, é possível visualizar o arquivo in.txt usado como input no comando de execução do cliente e o arquivo out2.txt usado como output, tendo ambos o mesmo conteúdo após a execução.

2) Comparando o código implementado no passo 1 com o código original da atividade prática 4 em termos de melhor utilização da rede e, consequentemente, tempo de execução, há alguma vantagem em utilizar o código novo? Justifique.

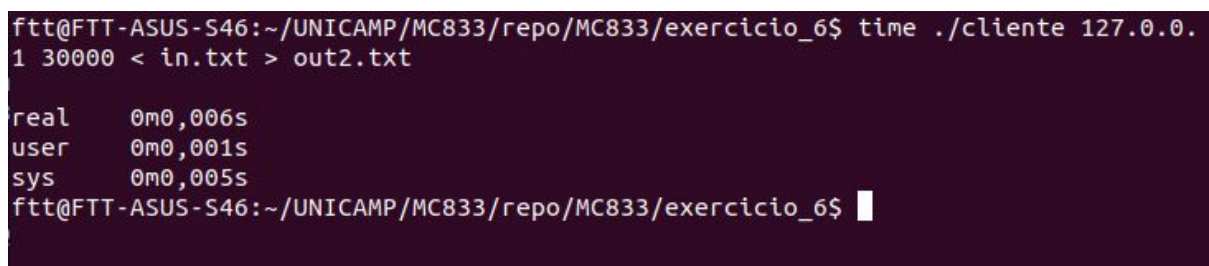


```
ftt@FTT-ASUS-S46: ~/UNICAMP/MC833/repo/MC833/exercicio_4/ex4_enviado
Arquivo Editar Ver Pesquisar Terminal Ajuda
ftt@FTT-ASUS-S46:~/UNICAMP/MC833/repo/MC833/exercicio_4/ex4_enviado$ ./servidor 30000 > server4out
sh: 1: oi: not found
sh: 1: a: not found
sh: 1: d: not found
sh: 1: c: not found
sh: 1: q: not found
sh: 1: q: not found
sh: 1: s: not found
sh: 1: s: not found
sh: 1: asfasfsafsafafafa: not found
sh: 1: d: not found
sh: 1: d: not found
sh: 1: d: not found
sh: 1: d: not found
sh: 1: d: not found
sh: 1: d: not found
sh: 1: d: not found
sh: 1: d: not found
sh: 1: d: not found
sh: 1: d: not found
Accept error: Bad file descriptor

```

```
ftt@FTT-ASUS-S46: ~/UNICAMP/MC833/repo/MC833/exercicio_4/ex4_enviado
Arquivo Editar Ver Pesquisar Terminal Ajuda
ftt@FTT-ASUS-S46:~/UNICAMP/MC833/repo/MC833/exercicio_4/ex4_enviado$ time ./cliente 127.0.0.1 30000 < in.txt > outclient
real    0m0,043s
user    0m0,003s
sys     0m0,003s
ftt@FTT-ASUS-S46:~/UNICAMP/MC833/repo/MC833/exercicio_4/ex4_enviado$
```

Execução do código do exercício 4, com tempo de 0,043s.



```
ftt@FTT-ASUS-S46:~/UNICAMP/MC833/repo/MC833/exercicio_6$ time ./cliente 127.0.0.1 30000 < in.txt > out2.txt
real    0m0,006s
user    0m0,001s
sys     0m0,005s
ftt@FTT-ASUS-S46:~/UNICAMP/MC833/repo/MC833/exercicio_6$
```

Execução do código deste exercício, com tempo de 0,006s.

Há uma significativa diferença de tempo de execução, com o código deste exercício em vantagem. Além da implementação com *select*, outra diferença a ser considerada é que o código do exercício 4 executa os comandos enviados pelo cliente no servidor, diferentemente deste exercício que apenas trata conjuntos de caracteres isoladamente.