

B121

Sistemas Operativos
Licenciatura em Engenharia Informática
Universidade do Minho

2013/2014

Duarte Duarte 61001

INTRODUÇÃO	3
B121	4
O porquê de B121	4
UTILIZAÇÃO	4
Inicialização do Cliente	4
Interação do Cliente	4
Formato cliente blank	5
Pedido de agregação gerou o ficheiro agregacoesBraga.txt	5
Reinício de um processo filho	6
FUNCIONAMENTO	6
CONCLUSÃO	8
DEMONSTRAÇÃO DO FUNCIONAMENTO DO SISTEMA SERVIDOR-CLIENTE	9

Introdução

Este projeto foi desenvolvido no âmbito da Unidade Curricular de Sistemas Operativos inserida na Licenciatura em Engenharia Informática. Com este projeto pretende-se desenvolver um sistema Cliente-Servidor em que se ponha em prática os conhecimentos adquiridos nesta UC ao longo do semestre.

É um trabalho que de certeza aborda toda os conhecimentos fulcrais e também prima por ser um projeto interessante fazendo com que o tempo dedicado a ele não seja visto como tempo de trabalho mas sim tempo investido e tempo de lazer.

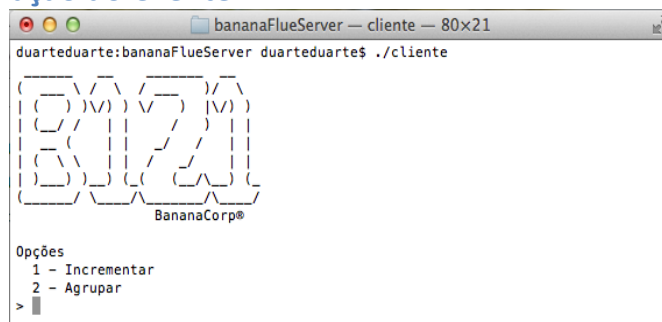
B121

O porquê de B121

Na fase inicial do trabalho havia uma função que estava sempre a dar um erro e na utilização do gdb o breakpoint era colocado como 'b 121'.

Utilização

Inicialização do Cliente



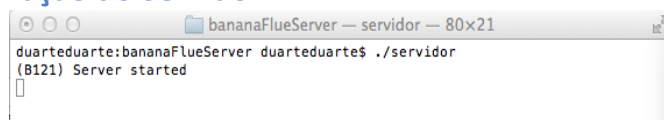
```
duarteduarte:bananaFlueServer duarteduarte$ ./cliente

BANANA
BananaCorp®

Opções
1 - Incrementar
2 - Agrupar
>
```

O programa Cliente é inicializado e tem este aspecto, esta é versão em que existe interação com o utilizador, mais à frente aparece uma versão em que não existe interação com aspecto “bonito” e com menu.

Inicialização do Servidor



```
duarteduarte:bananaFlueServer duarteduarte$ ./servidor
(B121) Server started
```

Quando o Servidor inicializa tem apenas uma mensagem a dizer que este foi inicializado.

Interação do Cliente



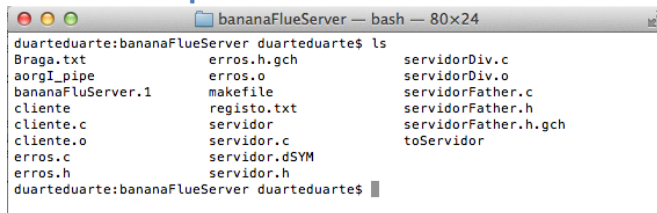
```
duarteduarte:bananaFlueServer duarteduarte$

BANANA
BananaCorp®

Opções
1 - Incrementar
2 - Agrupar
> 1
--Incrementar--
Indique a localização separada por virgulas
Localização: Braga,Braga,S.Vitor
Incremento: 10
(B121) Sent
Mon Jun 2 02:24:29 2014
duarteduarte:bananaFlueServer duarteduarte$
```

O Cliente preenche (no caso do menu e com interação) os dados que pretende caso seja incremento ou uma agregação e o programa envia a mensagem para o cliente, caso o cliente esta sobrelotado ou indisponível o cliente recebe um aviso a dizer que a mensagem não pode ser entregue.

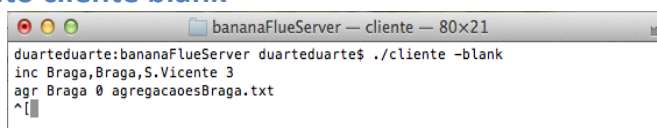
Objetos auxiliares que foram criados



```
duarteduarte:bananaFlueServer duarteduarte$ ls
Braga.txt          erros.h.gch        servidorDiv.c
aorgI_pipe         erros.o            servidorDiv.o
bananaFlueServer.1 makefile           servidorFather.c
cliente            registo.txt        servidorFather.h
cliente.c          servidor           servidorFather.h.gch
cliente.o          servidor.c         toServidor
erros.c            servidor.dSYM
erros.h            servidor.h
```

Foram criados vários “objetos” auxiliares, entre os quais o ‘aorgI_pipe’, o ‘toServidor’, o ‘registo.txt’ e ainda o ‘Braga.txt’, pelos menos estes são os que vale a pena frisar. O aorgI_pipe é um pipe com nome para o Cliente receber a resposta do Servidor. O toServidor é um pipe com nome para que todos os Clientes possam enviar mensagens para o Servidor. O registo.txt é um ficheiro que terá todo o histórico das mensagens que são enviadas para o Servidor. E por fim o Braga.txt foi criado porque a mensagem que foi enviada era dirigida ao distrito de Braga e o processo responsável por Braga guarda a informação das mensagens a ele dirigidas neste ficheiro.

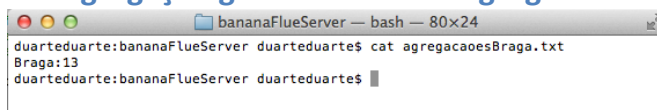
Formato cliente blank



```
duarteduarte:bananaFlueServer duarteduarte$ ./cliente -blank
inc Braga,Braga,S.Vicente 3
agr Braga 0 agregacoesBraga.txt
^I
```

O formato do cliente quando é inicializado com ‘-blank’ é um formato “em branco” em que apenas se põe a mensagem de acordo com o que se pretende obter do Servidor.

Pedido de agregação gerou o ficheiro agregacoesBraga.txt



```
duarteduarte:bananaFlueServer duarteduarte$ cat agregacoesBraga.txt
Braga:13
duarteduarte:bananaFlueServer duarteduarte$
```

Quando se pede uma agregação é dado o nome do ficheiro que conterá essa agregação. E neste caso (pela imagem acima vê-se um pedido de agregação e desde o inicio das imagens vê-se que foram incrementadas 13 vezes no distrito de Braga) pode-se ver que o ficheiro com o nome indicado tem as agregações do distrito de Braga.

Reinício de um processo filho

```
duarteduarte@bananaFlueServer duarteduarte$ ./cliente -blank
inc Braga,Braga,S.Vicente 3
agr Braga 0 agregacoesBraga.txt
inc Lisboa,Lisboa,Lumiar 19
```

Primeiro foi enviada uma nova mensagem desta vez destinada a outro distrito para que fosse criado mais um processo de forma a mostra o funcionamento do reinicio de processos.

```
2301 s002 S+ 0:00.00 ./servidor
2308 s002 S+ 0:00.00 ./servidor
2309 s002 S+ 0:00.00 ./servidor
2348 s002 S+ 0:00.00 ./servidor
duarteduarte@bananaFlueServer duarteduarte$ kill -SIGKILL 2348
```

Então para testar opta-se por “matar” o processo responsável por Lisboa.

```
duarteduarte@bananaFlueServer duarteduarte$ ./servidor
(B121) Server started
(B121) Waiting for connection
(B121) Message received
(B121) Message received
(B121) O filho de Lisboa esta a ser reiniciado
(B121) O filho de Lisboa esta novamente activo
```

O Servidor reagiu reiniciando o processo responsável por Lisboa.

Funcionamento

O servidor cria um processo para ter o registo global das mensagens, isto é, quando uma mensagem chega ao Servidor, seja destinada a que distrito for ela é enviada para por um pipe para o processo que faz os registos globais. Depois dessa mensagem é obtido o distrito ao qual ela se destina. Quando se sabe a que distrito ela pertence é então reencaminhada para o pipe do processo que trata desse distrito. Se ainda não houver um processo responsável por esse distrito é criado um novo. Cada processo responsável por um distrito regista num ficheiro próprio desse distrito as mensagens que recebe e para além disso trata de fazer incrementos e agregações.

Quando o processo “morre” através do SIGCHLD enviado para o pai o pai sabe que o processo morreu e então procede à sua reiniciação. Cria um novo processo que ocupará o lugar do processo que morreu. Quando esse novo processo é criado o pai atribui-lhe o pipe que era do processo que morreu e assim a comunicação fica exatamente como estava. Enquanto um processo está a reiniciar podem chegar mensagens destinadas a esse distrito e então o pai coloca as mensagens numa queue para serem entregues quando o processo estiver pronto. Quando o processo acaba de reiniciar, isto é, quando já recuperou o estado que tinha aquando da sua morte envia um SIGUSR1 ao processo pai para que este possa enviar as mensagens que estão na queue.

Na ultima página, é possível ver um esquema com o formato do funcionamento do sistema.

Conclusão

Com este trabalho foi-me permitido por em prática todas as ferramentas que adquiri ao longo de um semestre nesta UC. Foi um projeto que gostei bastante de realizar pois permitiu-me consolidar conhecimentos que pensei que estavam consolidados mas estavam apenas “decorados” por alto. Acho que depois de realizado posso dizer que contribuiu bastante para o que agora posso afirmar que sei.

As maiores dificuldades que senti ao longo deste projecto foi o facto do processo “pai” e os processos “filho” não partilharem o mesmo espaço de memória.

Demonstração do funcionamento do sistema Servidor-Cliente

S – servidor

S1 – processo que trata de tudo o que esteja relacionado com o distrito 1. (todos os Sn funcionam da mesma forma)

C1 – Cliente que se liga ao servidor para enviar uma mensagem relacionada com o distrito 1 (todos os Cn funcionam da mesma forma)

Retângulos pretos são pipes anónimos.
Retângulos brancos são pipes com nome.
Reg – processo responsável por registar todas as mensagens.
F1 – Ficheiro com a informação do distrito 1 (todos os Fn funcionam da mesma forma)

