

Guião 1

Discord da UC

<https://discord.gg/Evze2N36ST>

Usar o putty para aceder à máquina virtual

Vai receber um e-mail com as credenciais para aceder à máquina virtual no endereço `ud.di.uminho.pt`. Poderá aceder à máquina usando o putty no Windows. Consulte mais instruções aqui.

<https://web.di.uminho.pt/info2di/ssh2info/>

<https://www.ionos.com/digitalguide/server/configuration/use-ssh-keys-with-putty/>

Se tiver problemas com a chave veja se esta clarificação do nosso administrador não resolve o problema:

se estiver a usar um Linux por favor faça `dos2unix` da chave privada. Pelos vistos o correio eletrónico altera os "enter" de `\n` (linux/unix) para `\r\n` (windows) o que embora visualmente os ficheiros pareçam iguais são diferentes!

Vídeos a ver antes das aulas

<https://www.youtube.com/watch?v=qgszy9GquRs>

<https://www.youtube.com/watch?v=xSnetY3eolk>

Ficheiros

Os ficheiros dos guiões estão em `~rcm/guiões`

Podes criar uma "cópia local" escrevendo o seguinte na consola:

```
ln -s ~rcm/guiões ~
```

Editores que podes usar:

- Nano

- Vim

Objetivos

- Aprender a compilação separada
- Fazer uma makefile simples
- Correr um programa dentro do gdb
- Ser capaz de imprimir o conteúdo de variáveis
- Ser capaz de colocar um breakpoint numa linha e numa função
- Ser capaz de colocar um breakpoint condicional

Passos a executar

1. Vá buscar os dois ficheiros e coloque-os numa pasta;
2. Crie uma **makefile** para compilar o programa;
3. Use a opção **-ggdb** para compilar com informação para o debugger;
4. Use a flag **-lm** para compilar com a biblioteca de matemática;
5. A sua **makefile** deve suportar a compilação separada;
6. Corra o programa dentro do **gdb**;
7. Use o comando **help**;
8. Veja o que fazem os comandos **run** e **cont** no **gdb**;
9. Use o comando **help** para cada um dos comandos do **gdb** que encontrar abaixo;
10. Liste o conteúdo da função **main** e da função **raizes** usando o comando **list**;
11. Coloque um breakpoint na função **raizes** usando o comando **b**;
12. Corra o programa e escreva 6 6 -12;
13. Imprima o valor dos argumentos da função **raizes** usando o comando **p**;
14. Saia do gdb (em alternativa escreva veja a ajuda com **help breakpoints** para ver como apagar um breakpoint);
15. Volte a correr o programa dentro do **gdb** mas coloque um breakpoint na linha que imprime os valores com a função **printf**;
16. Imprima os valores das variáveis **r1** e **r2**;
17. Coloque um breakpoint na linha que atribui o valor à variável **r1**;
18. Corra o programa dentro **gdb** e introduza os valores 7 4 8;
19. Imprima o valor da variável **delta**, o que se passa?
20. Consulte a ajuda dos comandos **display** e **watch**;
21. Usando o comando **cond**, faça com que o programa pare no breakpoint só nos casos que identificou acima;
22. Volte a correr o programa dentro do **gdb** com os dois casos que testou acima e verifique que ele só pára quando é necessário.

Avaliação

- Ponto 15
- Ponto 20
- Ponto 22
- Ponto 23

Recursos

- Tutorial de makefiles:
https://web.stanford.edu/class/archive/cs/cs107/cs107.1174/guide_make.html
- Tutorial do gdb: <https://www.geeksforgeeks.org/gdb-step-by-step-introduction/>