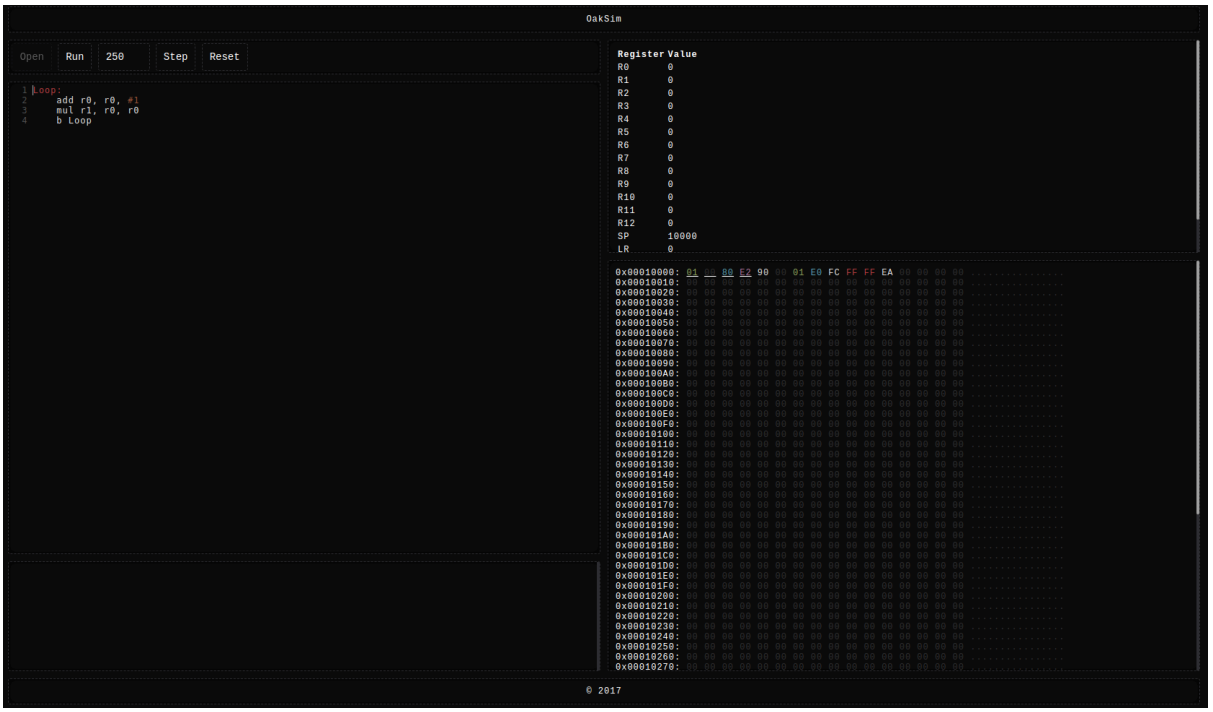
Prática 2

Parte 1: Linguagem Assembly

Nesta aula vamos aprender um pouco sobre linguagem assembly para a arquitetura de cpus ARM (advanced RISC machine). Para isso vamos usar um simulador online (disponível em https://wunkolo.github.io/OakSim/ ou https://www.fc.up.pt/pessoas/nuno.guimaraes/ii2425/arm64/).
Este simulador permite a escrita de código em linguagem assembly e a sua execução. O ambiente do simulador é mostrado na Figura abaixo



O simulador é composto de 3 janelas principais. Do lado esquerdo é o editor de código onde vamos escrever as instruções a serem executadas. Em cima do editor temos algumas opções como correr/parar o código (botão RUN), um valor numérico que determina o tempo de execução por cada instrução (500 significa que a execução de uma instrução vai demorar 500 ms), um botão STEP que permite uma execução passo-a-passo do código e um botão RESET que permite limpar todos os dados.

1) Corra o código de exemplo e veja o que acontece. Consegue determinar o que são a janela superior e inferior do lado direito?
2) Pare a execução clicando no botão RUN e faça reset. Agora, execute o código passo a passo e observe o que acontece na janela superior direita. Tente explicar o que fazem as seguintes instruções assim como o que são os seus argumentos (r0,r1, #1, Loop)
    a) add r0, r0, #1
    b) mul r1, r0, r0
    c) b Loop
3) Faça novamente Reset e corra o programa passo-a-passo novamente. No entanto, desta vez, observe o valor PC. Tente explicar o que este representa.

(só avançar após fazer a parte 1 pois contém respostas a algumas das questões)

Exercícios Extra

Vamos agora tentar realizar alguma programação muito simples em linguagem assembly. Para tal vamos precisar de perceber algumas instruções que são possíveis. Uma breve *cheat sheet* está disponível aqui (https://cheatography.com/syshella/cheat-sheets/arm-assembly/pdf/) ou na página da cadeira.

Tente escrever programas para as seguintes tarefas

4) Colocar o valor 5 no registo R2
5) Colocar a soma do registo R3 e R4 em R0. Inicialmente os registos R3 e R4 deverão ter os valores 4 e 2, respetivamente
6) Comparar os valores nos registos R2 e R3. Se o valor de R2 é maior então colocar o valor 1 em R0, caso contrário colocar o valor 0. (Pista: precisa das instruções beq e cmp). Teste ao colocar os valores 5 e 6 e 2 e 1 nos registos R2 e R3 respetivamente.
7) Multiplicar o valor presente no registo R2 com o valor no Registo R3 sem usar a operação de multiplicação e colocar em R0

8) Calcular a potência de um número no registo R2 com o exponente no registo R3 e colocar em R0

Parte 2: Terminal Linux

Nesta segunda parte vamos trabalhar com um terminal Linux. Como não existe uma versão do sistema operativo instalada nos computadores, vamos recorrer novamente a um simulador.
Para simular um terminal Linux vamos recorrer ao seguinte link https://bellard.org/jslinux/vm.html?url=alpine-x86.cfg&mem=192.

1. Execute os seguintes comandos e tente interpretar o que aparece no ecrã e que operação eles realizam. Para confirmar, pode consultar as informações sobre cada comando em https://linux.die.net/man/
    a. echo ola mundo
    b. date
    c. hostname
    d. arch
    e. uname -a
    f. uptime
    g. top (pode precisar de clicar em q para sair)↵
    h. env
    i. echo $HOME
    j. ps aux
    k. clear
    l. cal
    m. cal 2013
    n. cal 9 1752 (*alguma coisa de estranho ?*)
    o. history
2. Experimente agora os seguintes comandos que lhe vão permitir movimentar-se entre directórios.
    a. cd
    b. pwd (em que directório está?)
    c. ls -la (listagem de ficheiros)
    d. cd .
    e. pwd
    f. cd ..

g. pwd
h. ls -la
i. cd ..
j. pwd
k. ls -la
l. cd ..
m. ls -la
n. pwd
o. cd /etc
p. ls -la
q. ls -la | more
r. cd
s. pwd
t. cd teste
u. mkdir teste (criar directório)
v. cd teste
w. ls -la
x. echo ola mundo > ola.txt
y. ls -la
z. cat ola.txt (mostrar conteúdo de um ficheiro)
aa. cp ola.txt copia.txt (copiar ficheiro)
bb. ls -la
cc. cat copia.txt
dd. rm copia.txt (remover ficheiro)
ee. ls -la
ff. rm /bin/ls (o que acontece?)

3. Use os comandos cd, ls, pwd e cat para explorar o directório /bin
4. Faça a seguinte sequência de comandos e veja se percebe o que vai acontecendo e o que significa o último comando.
    a. history
    b. history | awk '{print $2 }'
    c. history | awk '{print $2 }'| sort
    d. history | awk '{print $2 }'| sort | uniq -c
    e. history | awk '{print $2 }'| sort | uniq -c | sort -nr
    f. history | awk '{print $2 }'| sort | uniq -c | sort -nr | head -5

Exercícios Extra

5. Vamos agora verificar a utilidade de usar o terminal em tarefas repetitivas.
    a. bash
    b. cd
    c. mkdir testes
    d. cd testes
    e. for ((i=1; i <=100; i++)) do mkdir dir_$i; done (criar 100 diretórios)
    f. ls * (listar o conteúdo dos 100 diretórios - poderá tentar verificar os 100 diretórios através do ambiente gráfico)
    g. ls dir_8* (o que é que acontece?)
    h. ls dir_*8 (o que é que acontece?)
    i. for ((i=1; i <=100; i++)) do rmdir dir_$i; done (remover 100 diretórios)