Módulo 9
Auto-Vectorização

Copie o ficheiro `/share/acomp/LOOP-P09.zip` para a sua directoria, faça o unzip e gere os executáveis com nível de optimização O2 e 2 opções diferentes: **`novec-kernel`** não inclui vectorização, enquanto **`vec-kernel`**inclui a opção de auto-vectorização pelo compilador. Para esse efeito, basta escrever **`./mymake.sh`**

Este comando gerará também os ficheiros **`vec-kernel.s`** e **`novec-kernel.s`** contendo o *assembly* correspondente aos diferentes níveis de optimização.

Ao longo deste módulo iremos concentrar a nossa atenção nas funções **`loop1() a loop6(), loop_AoS()`** e **`loop_SoA()`** que se encontram em **`vec-kernel.c`**.

Exercício 1 - Para cada uma das funções **`loop1()`** a **`loop6()`** e **`loop_AoS()`** examine o respectivo código e indique justificando, na coluna "Obs." da Tabela 1, se pensa tratar-se, ou não, do código vectorizável pelo compilador. As linhas correspondentes a **`loop3-V2()`** e **`loop_SoA()`** serão preenchidas nos exercícios posteriores.

Preencha a para ambas as versões, cuja principal diferença será a vectorização. Note que pode verificar a vectorização consultando os ficheiros **`.s`**. Para executar cada versão da função **`loop()`** use o comando **`qsub`** conforme indicado abaixo para a versão **`loop1()`**. Este comando executará ambos os executáveis, isto é, **`novec-kernel e vec-kernel.`** O argumento usado para selecionar a versão pode ser consultado na coluna **`V`** da Tabela 1:

```
qsub –F "1" loop.sh
```

Exercício 2 - Analize a função **`loop3().`** Porque é que não vectoriza? Consegue sugerir uma alteração ao código desta função de forma que vectorize? Se sim, altere-a e preencha a linha **`loop3-V2()`** acima.

Exercício 3 - Escreva a função **`loop_SoA(),`** para que realize os mesmos cálculos que **`loop_AoS()`** mas usando uma estrutura de *arrays*.

Note que:

- o tipo de dados **`My_SoA`** já está definido em `vec-kernel.h`:
  `typedef struct {float *a, *c} MY_SoA;`
- a variável **`SoA`** é declarada em `main.h` e os respectivos vectores (`a` e `c`) devidamente inicializados.

Exercício Complementar

Considere a função **`loop5(),`**que não vectoriza devido a uma dependência RAW. Consegue reorganizar as computações de forma a que esta possa vectorizar parcialmente?
Sugestão: Pode separar o ciclo em dois ciclos, um deles sem dependências!

func	V	opt	Texec (ms)	#I (M)	CPI	#VEC_SP (M)	Obs.
loop1()	1	no					
		vec					
loop2()	2	no					
		vec					
loop3()	3	no					
		vec					
loop3-V2()	3	no					
		vec					
loop4()	4	no					
		vec					
loop5()	5	no					
		vec					
loop6()	6	no					
		vec					
loop_AoS()	7	no					
		vec					
loop_SoA()	8	no					
		vec					

Tabela 1