

Compilação e *Debugging* em Unix

1. Copie os ficheiros da página da disciplina para a sua área de trabalho e visualize-os com o editor de texto favorito de modo a compreender o respectivo código. Altere o ficheiro `fact.h` para que a dimensão do vector (`DIM_DADOS`) seja 1.
2. Compile os ficheiros fonte e faça a sua ligação de modo a produzir o executável respectivo denominado `fact`.

```
$ gcc -g -c fact.c fact_iter.c fact_rekurs.c  
$ gcc fact.o fact_iter.o fact_rekurs.o -lpthread -g -o fact
```

3. Execute o programa e verifique que este preenche um vector com um elemento inteiro aleatório e lança uma tarefa onde é feito o cálculo do factorial do inteiro contido na primeira posição do `array`.
4. Crie uma `Makefile` para automatizar o processo de criação do executável `fact`. Inclua também um *target* **clean** que permita remover todos os ficheiros `.o` e o executável.
5. Verifique que o programa `make` faz a actualização do código como previsto.

```
$ make clean  
$ make  
$ ./fact
```

6. Execute o programa `fact` dentro do *debugger* `gdb` e visualize o código fonte.

```
$ gdb fact  
<gdb> list
```

7. Recorrendo ao *debugger*, insira um *breakpoint* após a inicialização do vector `dados`, execute-o, e modifique a variável para ter `dados[0]=5`. Continue a execução. Que observa?

```
<gdb> break 35  
<gdb> run  
<gdb> set dados[0]=5  
<gdb> continue
```

8. Use o comando `break` do `gdb` para parar a execução do programa em cada iteração do ciclo `for` da função `factorial_iterativo`. Observe os valores de `n` e do total acumulado na variável `factorial` em cada iteração.

```
<gdb> break fact_iterativo  
<gdb> list fact_iterativo  
<gdb> break fact_iter.c:19  
<gdb> display factorial  
<gdb> continue
```

Atenção: o comando `display factorial` deve ser feito quando o programa se encontrar parado no breakpoint da rotina `fact_iterativo`.

9. O ficheiro `fact_rekurs.c` contém uma rotina que se encontra vazia. Codifique-a de acordo com a sua descrição no cabeçalho. Modifique o ficheiro `fact.c` de forma a que o programa passe a executar a rotina que calcula o factorial de forma recursiva. Execute o programa `fact` modificado dentro do *debugger* e verifique que funciona correctamente para `n = 0, 1, 2, 5`. Ponha um *breakpoint* na função recursiva e use os comandos `backtrace` e `frame` para observar as variáveis nos vários níveis de recursividade.
10. Aumente a dimensão do vector e recompile o programa (`make`). Observe a concorrência entre as várias *threads* executando o programa dentro e fora do `gdb`.