

# Avaliação de Desempenho

## Teste 5

2. Considere a optimização do código que calcula o factorial de um inteiro. Os 2 algoritmos mais populares para efectuar este cálculo baseiam-se na recursividade e num processo iterativo. Para efeitos de medição de desempenho, vamos adaptar e adoptar o conceito de CPE a este caso, uma vez que poderemos considerar que o nº de iterações está relacionado com o inteiro usado no cálculo do factorial. Usou-se então o seguinte algoritmo recursivo para o cálculo do factorial:

```
int fact1 (int n)
{
    int result=1;
    if (n>1) result = n * fact1(n-1);
    return result;
}
```

e o valor de CPE medido no cálculo de factoriais de 1000 a 5000 rondou o valor 50 (num P6).

Usando agora o seguinte algoritmo iterativo:

```
int fact2 (int n)
{
    int i, result=1;
    for (i=n; i>0; i--) {
        result = result*i;
    }
    return result;
}
```

o valor de CPE baixou para pouco mais que 4.

- a) <sup>(R)</sup> Identifique e caracterize as principais optimizações que se fizeram sentir com a transformação do algoritmo de recursivo em iterativo (uma melhoria de 50 para 4!).

- b) <sup>(R/B)</sup> Um programador entendeu que poderia fazer ainda melhor se modificasse o código para:

```
int fact3 (int n)
{
    int i, result=1;
    for (i=n; i>1; i-=2) {
        result = (result*i) * (i-1);
    }
    return result;
}
```

Nº	Nome
----	------

Caracterize sumariamente esta técnica de optimização.

- c) (B) As medições de desempenho que foram feitas com esta técnica apresentaram valores de CPE sem qualquer melhoria em relação à versão anterior (iterativa). Como explica este resultado?

- d) (B) Se alterar a linha de código no interior do ciclo iterativo para:

```
result = result* (i * (i-1))
```

os valores medidos de CPE andam próximos de 2.5.

Como explica esta melhoria significativa no desempenho deste código?

Sugestão: represente o diagrama temporal de execução referente a 3 iterações e tente explicar com base no comportamento observado. Se ajuda, segue-se o código *assembly* do interior do ciclo para a 1ª versão iterativa, para a versão do programador e para esta última modificação (por esta ordem).

<pre>L16:   imull %edx, %eax   decl  %edx   testl %edx, %edx   jg    L16</pre>	<pre>L24:   imull %edx, %ecx   leal  -1(%edx), %eax   imull %eax, %ecx   subl  \$2, %edx   cmpl  \$1, %edx   jg    L24</pre>	<pre>L32:   leal  -1(%edx), %eax   imull %edx, %eax   imull %eax, %ecx   subl  \$2, %edx   cmpl  \$1, %edx   jg    L32</pre>
--	--	--

Nº	Nome
----	------