

00 Experimente a criação de processos que compartilham o espaço de endereçamento de memória. Analise o resultado

```
1  #include <stdio.h>
2  #include <unistd.h>
3
4  int main() {
5      pid_t pid;
6      pid = fork();
7      if (pid == 0) {
8          printf("Este é o processo filho, ID do processo: %d\n", getpid());
9      }
10     else {
11         printf("Este é o processo pai, ID do processo filho: %d\n", pid);
12     }
13     return 0;
14 }
```

01 Escreva três funções em C ou C++: uma que declare um grande vetor estaticamente, outra que declare o mesmo grande vetor na pilha (enviando o vetor como parâmetro de uma função) e outra que crie o mesmo grande vetor no monte (criando o vetor dentro da função). Chame cada um desses subprogramas um grande número de vezes (ao menos 100 mil vezes) e mostre na tela o resultado necessário para cada um. Explique os resultados.

02 Escreva programas de teste em C++, Java e Python para determinar o escopo de uma variável declarada em uma sentença **for**. Especificamente, o código deve determinar se tal variável é visível após o corpo da sentença **for**.

03 Crie programas para mostrar escopo de variáveis através de chamadas a funções.

04 Exemplo de código em python para usar funções.
Experimente o uso da recursividade. Refaça o programa em C ou C++ e analise os resultados.

```
1 def saudacao1(nome):
2     print ("Olá, " + nome + "!")
3     saudacao2(nome)
4     print ("Preparando para dizer tchau...")
5     tchau()
6
7 def saudacao2(nome):
8     print ("Como vai " + nome + "?")
9
10 def tchau():
11     print ("Ok, tchau!")
12
13 def main():
14     saudacao1("lucilia")
15
16 if __name__ == "__main__":
17     main()
```

```
1 def fat(x):
2     if x == 1:
3         return 1
4     else:
5         return x * fat(x - 1)
6
7
8 def fatorial(n):
9     resultado = 1
10    for i in range(1, n + 1):
11        resultado *= i
12    return resultado
13
14 numero = 3
15 resultado = fatorial(numero)
16 print(f'O fatorial de {numero} é {resultado} (sem recursividade)')
17 resultado = fat(numero)
18 print(f'O fatorial de {numero} é {resultado} - recursividade')
```