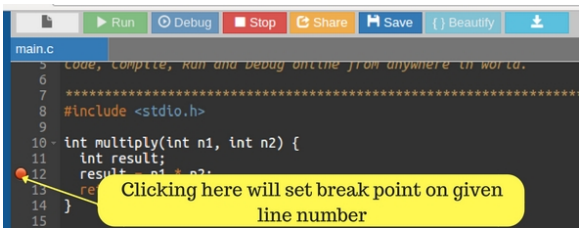
UFOP-DECOM-BCC264 Nº 03/2021-1

3º TP 2021-1

Veja o vídeo que fiz a respeito deste TP, ok? Este TP diz respeito a gerenciamento de memória.

Você fará uso do capítulo 7 do livro *Hands-On SystemProgramming with C++,* disponível no Moodle.

Para rodar os programas do livro, é necessário usar C++17 como linguagem (versões inferiores a 17, provavelmente não funcionará) e onlinegdb.com é muito interessante. Esta ferramenta tem um debug online, mas se faz necessário ler o tutorial, mas é muitíssimo simples. Sendo necessário, basicamente setar os breakpoints.



Mas, você NÃO precisa enviar nenhum código neste TP para mim. Mas vc pode rodar e depurar o programa que está no final deste documento.

Depois de entender e ler o programa e com base no capítulo 7 do livro acima, marque um "X" no local (heap ou no stack ou no global .bss ou global .data?) onde a variável foi criada e justifique sua escolha no campo "por quê?") na tabela abaixo.

	Heap	Stack	Global .bss	Global .data	Por quê?
`hypercounter`					
`counter`					
`*idk2`					
`Cook1`					
`clk1`					
`cook2->c`					
`*arr em clk1`					
`*arr em clkk2`					

Tabela 1

Pois é, temos o *heap memory* que, neste caso, significa "free memory", e a pilha (stack memory). Além disto temos as memórias globais (Global memory) que são localizados em duas diferentes localidades denominadas .bss e .data.

Existe um sinergia muito importante entre o compilador, o linkeditor e OS-Loader. Veja as 4 primeiras páginas do capítulo acima referenciado.

No https://en.wikipedia.org/wiki/Memory_management temos um overview de gerenciamento de memória que fala sobre:

Fragmentação Externa e eficiência;
E quatro tipos de implementações:

1. Fixed-size blocks
2. Buddy blocks
3. Slab Allocation
4. Stack Allocation

No capitulo 7 do livro acima, pagina 220, .. existe uma seção sobre alinhamento de memória que pode se feito de 3 formas:

1. Globalmente
2. Na Pilha
3. Dinamicamente

O Nosso TP você explicar os conceitos que estão em highlighting (amarelo) e completar a tabela 1

Bom, o nosso TP não abrange todo o restante do capitulo 7 mas não quer dizer que o mesmo não trata de questões interessantes como tratamento de exceções (`throw` exceptions), alinhamento de variáveis criadas com o operador `new`() e questões de alocação de memória em operações de *overloading* (lembra-se de OOP? C++ e todas outras implementam operações de *overloading*), entre outras coisas.

O que deve entregue:

Entregáveis:

a. 1 texto explicando as coisas acima.
b. O vídeo e 3 minutos onde você é o professor explicando sobre as questões acima.. 3 minutos é como um pitch. NÃO LEIA nenhum texto (veja abaixo)
c. POR FAVOR, NÃO LEIA O TEXTO, me explique o que fez.. Se sua apresentação for ler o texto (o que vc apresentou ou outro), será desconsiderado.
d. arquivo em formato PDF de texto explicando o que foi feito;

POR FAVOR, NÃO ZIPE


```
#include <iostream>

#define p(s) std::cout << s << std::endl

int hypercounter =0;

int counter;

class Cook
{
public:
    char c = 'c';  //
    Cook() {
        p(c);
    }
};

class CLK
{
public:
    int* arr = new int[1]; //
    CLK() {
        Cook Cook3;        //
        arr[0] = 90;
        Cook* Cook4 = new Cook(); //
        p(arr[0]);
    }
};

int main(int argc, char const *argv[])
{
    Cook Cook1;
    Cook* Cook2 = new Cook();

    CLK clk1;
    CLK* clk2 = new CLK();

    return 0;
}
```

(oculto em pdf ,apenas word)