

## **Atividades Aula 2**

### *Fluxo de Projeto*

**1)**

**a)** Um compilador traduz a linguagem de alto nível para uma de baixo nível (Assembly). Já o Assembler (montador) pega esse arquivo anteriormente compilado e gera o Object File (arquivo “.O”). E por fim o Linker (ligador) gera o arquivo executável final, ou seja, pega os arquivos “.O” e gera o arquivo que será executado ou interpretado pela máquina.

**b)** É um Sistema Operacional que controla o tempo de execução de cada tarefa. Por esse motivo é determinado RTOS, ou seja. Sistema Operacional de Tempo Real. É utilizado em sistemas onde haja a necessidade de controle de tempo de execução de cada tarefa do SO.

**c)** O Modelo V é um modelo conceitual de Engenharia de Desenvolvimento de Produto visto como melhoria ao problema de reatividade do modelo em cascata. Ele permite que, durante a integração de um sistema em seus diversos níveis, os testes sejam feitos contra os próprios requisitos do componente que está sendo testado, em contraste com modelos anteriores onde o componente era testado contra a especificação. Principais características:

- Enfatiza a importância de considerar as atividades de testes durante o processo, ao invés de um teste posterior após o término do processo;
- Pode-se obter a retroalimentação mais rapidamente;
- Ajuda a desenvolver novos requisitos;
- Melhora a qualidade do produto resultante.

Desvantagens:

- Dificuldade em seguir o fluxo sequencial do modelo;
- Dificuldade para o cliente poder especificar os requisitos explicitamente.

**d)** A principal diferença entre em linguagem C e C++ é o fato desse segundo ser orientado a objetos e o primeiro é uma linguagem procedural.

**2.5)** Em C, a ordem das funções importa no momento da compilação e por esse motivo função main deve ser a última em um arquivo “.c”. Porém em vários momentos é necessário dividir um determinado arquivo em vários outros, montando assim uma biblioteca.

Então de forma a organizar e evitar memorizações desnecessárias faz uso dos protótipos de funções que ficam localizados antes das demais funções e no início de um arquivo “.h”.

Essa questão resolve o problema da ordenação do código, organiza nosso código e informa (no arquivo header “.h”) quantas e quais são as funções dispostas em nossa biblioteca.

**2.6)** `count = 20, *temp = &sum, sum = 20`

**2.8)** Um vetor (array) é um ponteiro, portanto as duas formas estão corretas.

**2.9)** O único parâmetro de entrada é a variável inteira *teste\_array*, deve ser desta maneira: *int print\_array (int teste\_array[20])*.

**2.10)** Para retornar mais de um valor deve-se declarar explicitamente um ponteiro e retornar apenas o ponteiro.

**2.11)** A função *malloc(n)* libera n bytes de memória e retorna o endereço da primeira posição desse byte. Após a utilização essa memória deve ser liberada. Para isso utiliza-se a função *free()*.