REVISÃO PARA A PROVA 03 – PROGRAMAÇÃO DE SISTEMAS
CURSO DE CIÊNCIA DA COMPUTAÇÃO
UNIVERSIDADE FRANCISCANA – UFN. 2025-02.
Peso:2,0.

PROFESSOR: André F. dos Santos.
Nome do aluno: Pedro Henrique de Brito Canabarro. Data: 23/10/2025.

Destaque em amarelo a alternativa correta nas questões de múltipla escolha. Preencha seu nome e data. Questões de múltipla escolha (marque apenas uma alternativa).

1) Em um fluxo típico de construção (build) de um programa em C, qual é a diferença principal entre compilação/montagem, ligação (link-edit) e carregamento (load)?
a) Compilação resolve referências entre módulos; ligação traduz código-fonte para objeto; carregamento gera código de máquina.
b) Compilação e montagem geram arquivos-objeto; a ligação resolve símbolos e produz o executável; o carregamento mapeia o executável na memória e aplica a relocação final quando necessário.
c) Compilação coloca o programa diretamente na memória; ligação apenas cria uma tabela de símbolos; carregamento não altera endereços.
d) Compilação e ligação são a mesma etapa; carregamento apenas inicia o processo.
e) Compilação resolve endereços absolutos; ligação ignora símbolos externos; carregamento corrige o código-fonte.

2) Sobre símbolos e tabelas de símbolos no processo de ligação, assinale a correta: a) Símbolos definidos e não utilizados impedem a geração do executável.
b) Bibliotecas estáticas não contêm símbolos; só código binário puro.
c) Símbolos locais sempre são exportados para outros módulos.
d) O ligador ignora nomes de símbolos; apenas compara tamanhos.
e) Símbolos indefinidos são resolvidos pelo ligador cruzando definições em outros módulos ou bibliotecas.

3) Relocação é necessária porque:
a) O código-fonte não pode conter variáveis.
b) O endereço base do programa na memória é fixo e conhecido na compilação.
c) Endereços no código/ dados precisam ser ajustados para o endereço de carregamento final do programa ou para combinar módulos entre si.
d) O SO sempre executa em modo interpretado exigindo correção dinâmica.
e) Aumentamos a velocidade de execução do programa em 50%.

4) Sobre ligação estática versus dinâmica:
a) Na ligação estática, as bibliotecas são vinculadas apenas quando o programa começa a rodar.
b) Ligação dinâmica impede atualizar bibliotecas sem recompilar o aplicativo.
c) Ligação estática sempre reduz o tamanho do executável em relação à dinâmica.
d) Na ligação dinâmica, o programa não precisa de nenhuma biblioteca instalada no sistema para executar.

e) Ligação dinâmica permite atualizar bibliotecas compartilhadas sem recompilar o aplicativo, desde que sejam compatíveis.

5) Em formatos como ELF, as entradas de relocação (relocation entries):

a) São aplicadas pelo compilador e não chegam ao arquivo objeto.
b) Indicam posições no código/dados a serem ajustadas com base em símbolos e tipos de realocação.
c) Apenas existem para variáveis globais, nunca para funções.
d) São ignoradas no carregamento dinâmico.
e) Tornam desnecessária qualquer tabela de símbolos.

6) Explique o papel do ligador (linker) no processo de construção de um executável. Comente: resolução de símbolos, união de módulos-objeto, endereçamento, geração de seções e produção do binário final.

O ligador é responsável por pegar um ou mais módulos-objeto (gerados pelo compilador) e combiná-los em um único executável. Suas funções centrais incluem:

- **União de módulos e geração de seções:** Agrega seções de mesmo tipo (como .text para código ou .data para dados) de todos os arquivos de entrada em seções únicas no arquivo final.
- **Resolução de símbolos:** Encontra as definições para símbolos que são referenciados mas não definidos dentro de um módulo (ex: uma chamada à função printf definida em outro lugar).
- **Endereçamento (Relocação):** Ajusta os endereços no código e nos dados para que reflitam suas posições finais na memória, após todos os módulos serem combinados.
- **Produção do binário final:** Gera o arquivo executável completo.

7) Compare ligação estática e dinâmica, apontando vantagens e desvantagens de cada abordagem em termos de tamanho de binário, desempenho, atualização/patch, portabilidade e dependências em tempo de execução.

Ligação Estática: O código das bibliotecas é copiado para dentro do executável final.
Vantagens: O executável torna-se autossuficiente (melhor portabilidade, sem dependências externas em tempo de execução).
Desvantagens: Gera binários maiores (código duplicado em vários programas) e dificulta atualizações (se a biblioteca precisar de um patch, todos os programas que a usam devem ser recompilados).
Ligação Dinâmica: O executável contém apenas referências às bibliotecas (que são arquivos separados, como .so ou .dll).

Vantagens: Binários menores; permite que bibliotecas sejam atualizadas (patches) sem precisar recompilar os aplicativos; economiza memória, pois a biblioteca é carregada uma vez e compartilhada por vários processos.
Desvantagens: O programa depende que a biblioteca correta esteja instalada no sistema (dependências em tempo de execução).

8) Qual é o objetivo principal do processo de relocação durante o carregamento de um programa? a) Traduzir o código-fonte para linguagem de máquina.
b) Ajustar comentários do código para o novo endereço de memória.
c) Ajustar endereços no código e nos dados para o endereço em que o programa foi carregado.
d) Converter variáveis globais em locais para melhorar segurança.
e) Organizar funções em ordem alfabética na memória.

9) Em um erro de "símbolo indefinido" na etapa de link, qual é a causa mais comum? a) O compilador não gerou código de máquina.
b) O ligador não encontrou a definição do símbolo em nenhum módulo-objeto ou biblioteca incluídos na linkagem.
c) O sistema operacional bloqueou a execução por permissão.
d) O código-fonte tem uma variável global duplicada.
e) A CPU não suporta instruções de ponto flutuante.

10) Para que servem as entradas de relocação em um arquivo-objeto? a) Para remover comentários do código-fonte.
b) Para indicar posições no código/dados que precisam ter endereços ajustados quando o módulo for ligado ou carregado.
c) Para comprimir o executável final.
d) Para ordenar funções por tamanho.
e) Para impedir que o programa use bibliotecas externas.

11) Faça uma questão própria (de marcar ou responder) no assunto e coloque o gabarito.

Durante o processo de ligação, se dois módulos-objeto (main.o e utils.o) definem uma variável global com o mesmo nome (ex: int global_count), qual erro o ligador (linker) irá reportar?

a) Símbolo indefinido (Undefined symbol)
b) Erro de relocação (Relocation error)
c) Definição múltipla de símbolo (Multiple definition of symbol)
d) Violação de segmento (Segmentation fault)

Resposta: O ligador é responsável por resolver símbolos. Um erro de "símbolo indefinido" ocorre quando uma referência a um símbolo não pode ser encontrada em nenhum módulo ou biblioteca. Em contrapartida, um erro de "definição múltipla" ocorre quando o ligador encontra *mais de uma* definição para o mesmo símbolo global, tornando impossível para ele decidir qual endereço usar para as referências.