

# Compressão de código executável

# Agenda

- Compressão de dados
- Compressão de código
- Técnicas de compressão

# Compressão de dados

- Remover redundância nos dados de entrada
  - Repetição de símbolos
  - Alinhamento e *padding*
  - ...
- Compressão de texto
  - Dicionário
  - Estatístico
- Características
  - Adaptativo
  - Com ou sem perda

# Compressão de código

- Finalidade da técnica:
  - Redução do binário para armazenamento ou transmissão
  - Redução do binário em memória
- Momento da descompressão:
  - Tempo de execução (aleatória e com restrições)
  - Tempo de carregamento, transmissão, etc.
- Considerações:
  - Complexidade do descompressor
  - Tipo de descompressor (*software* ou *hardware*)

# Compressão de código

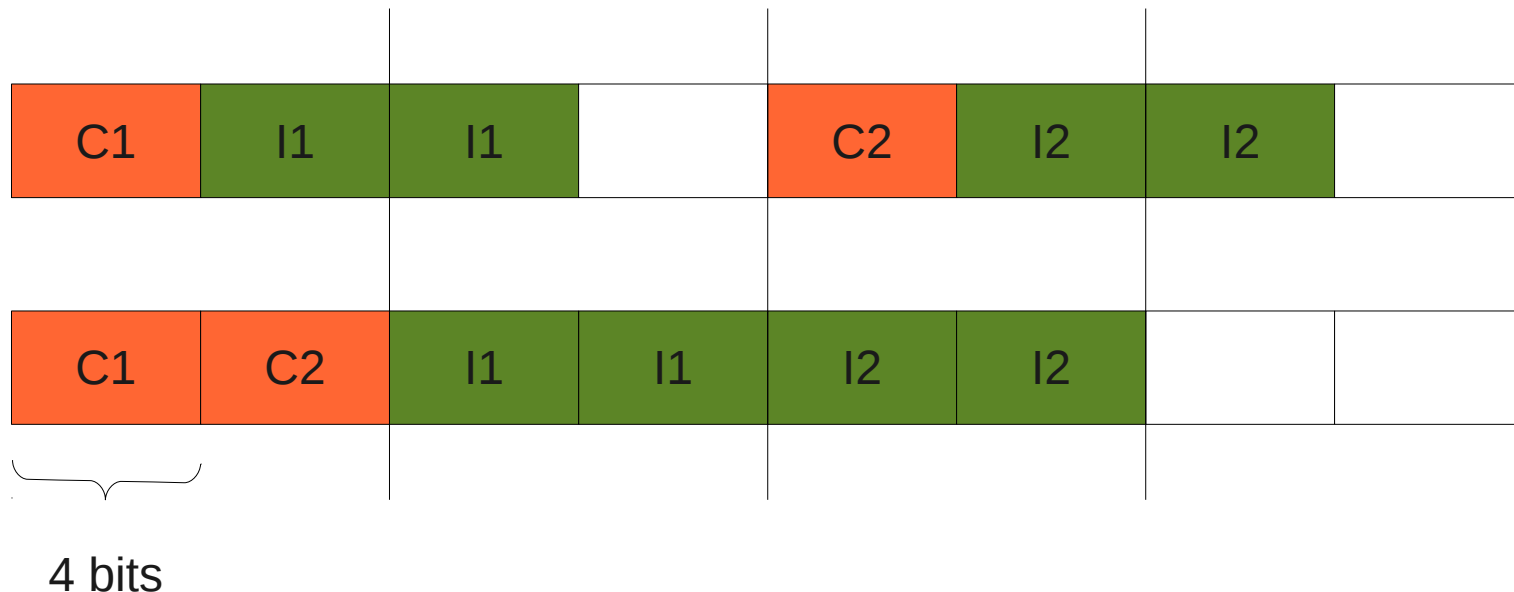
- Métodos de compressão mais comuns:
  - Compressão por dicionário
  - Empacotamento (*data packing*)

# Compressão de código

- Compressão por dicionário
  - Utiliza um dicionário com símbolos comuns
    - Partes de uma instrução
    - Instruções inteiras
    - Conjunto de instruções
  - Símbolos do programa são substituídos por índices
  - Melhoria:
    - Manter símbolos menos frequentes

# Compressão de código

- Empacotamento (*data packing*)
  - Reordenação para alinhamento à palavra
  - Melhor aproveitamento da memória
  - Melhor desempenho



# Técnicas de compressão

- Descompressão de instruções individuais
  - Thumb e MIPS16; DEL
- Descompressão na chamada de procedimento
  - Kirovski *et al*
- Descompressão em memória
  - *Compressed Code RISC Processor (CCRP)*
- Descompressão em tempo de carregamento
  - *Slim binaries*
- Compactação em tempo de compilação
  - Abstração de procedimentos; Otimização MMA



# Thumb e MIPS16

- Subconjunto de ARM e MIPS-III (ambos *32bits*)
- Instruções codificadas em 16 bits
  - Menos bits para registradores e imediatos
  - Durante a execução, mapeia instruções para correspondentes em *32bits*
- Requer mudança no modo de execução

# Directly Executed Languages (DEL)

- Representação intermediária do programa:
  - Utiliza operadores da linguagem de alto nível
  - Referencia diretamente os elementos (variáveis, etc.) como na linguagem de alto nível
  - A codificação considera características do programa

# Kirovski *et al*

- Procedimentos comprimidos com Ziv-Lempel
- Dois elementos:
  - **Serviço de diretório:** mapeia procedimentos entre o espaço de memória comprimido e descomprimido
  - **Cache de procedimento:** local reservado para os procedimentos descomprimidos
- Não requer modificações no processador
- Pode aumentar muito o custo das chamadas de procedimento

# CCRP

- Compressão de linhas de cache com Huffman
- Descompressão das linhas em tempo de execução
  - Quando ocorre cache miss, mais linhas são obtidas e descomprimidas da memória.
- Não requer modificações no núcleo do processador

# *Slim binaries*

- Formato independente de máquina
  - Representação comprimida da árvore sintática
- Descompressão durante o carregamento
  - Geração de código da árvore sintática
  - Permite geração de código durante execução

# Abstração de procedimentos

- Transforma padrões de instruções em procedimentos
- *Overhead* de chamadas de procedimento
- Liao *et al* propõem macro-subrotinas:
  - Não requer salvamento dos registradores
  - Não requer/permite passagem de parâmetros

# Otimização MMA

- Instruções *Multiple Memory Access* (MMA)
  - Permitem efetuar *load* e *store* sobre múltiplos dados
- Permite otimizações ao operar sobre vetores e estruturas.
- SolveMMA
  - Otimiza o código para explorar MMA
  - Reordena variáveis e otimiza operações sobre vetores

Muito obrigado!