

LAB 3 - HISTOGRAMA

Sistemas Embarcados - ELF74

Data: 05/10/2022

Alunos: Gabriel Bento Villanti e Rafael Pinheiro

1 - Planejamento das fases do processo de desenvolvimento

O planejamento deste relatório consistiu em compreender inicialmente o domínio do problema a forma como seria manipulada a imagem. Fizemos o estudo de como se dá a composição de um bitmap, visto que são pixels dispostos de forma matricial. A fase seguinte, se deu na compreensão de um histograma, utilizando dos conhecimentos adquiridos em sala e pesquisas para determinar a forma correta de atingir os valores de pixels desejados (neste caso de 0 a 255).

2. Definição do problema a ser resolvido

Feito o planejamento, partimos para a implementação do código em assembly que faz a construção do histograma da imagem que foi disponibilizada na proposta da atividade. Levando em conta que o retorno seria um sinal de 16 bits e as demais restrições do projeto tivemos o cuidado de tratar os valores corretamente para se obter sucesso no desenvolvimento do projeto.

Isso deve ser feito de maneira que seja respeitada a passagem dos argumentos e o retorno de valores do código.

3. Especificação da solução

Consiste na implementação de um código em C que chama uma função implementada em assembly que fará a construção de um histograma. Para isso será necessário utilizar o método AAPCS – ARM Architecture Procedure Call Standard para fazer a interface entre C e Assembly.

4. Estudo da plataforma de HW (Placa Tiva e seu processador)

Utilizamos os vídeos do professor do classroom para estudar mais sobre nosso processador e da placa em si. Também foi necessário estudar como as chamadas de função, através da utilização livre dos registradores R0 a R3 e R12, já os demais registradores para serem utilizados precisam ser salvos.

5. Projeto (design) da solução

5.1. Planejamento das estruturas de dados

5.2. Forma de passagem de parâmetros

Foram utilizados os registradores de R0 a R3 para passar 4 argumentos. Caso mais fossem necessários, então utiliza-se a pilha.

5.3. Algoritmo e alocação de variáveis aos registradores

6. Configuração do projeto no IDE (IAR)

A organização dos arquivos no projeto se deu de maneira que o arquivo “main.cpp”, o programa principal, fazia a chamada da função *EightBitHistogram* e criava os casos de teste que foram utilizados para validar o código e apresentar os resultados pelo terminal.

No arquivo “histogram.s” está contido o código em assembly que faz a construção do histograma. Nele estão descritas as rotinas “InicializaHistograma” que preenche o histograma com zeros e verifica se os 256 bits foram atingidos. Logo após a rotina “PreencheHistograma” que possui instruções para ler os valores do vetor e comparar até que se atinja o final da imagem. Por fim a rotina “EightBitHistogram” que é a principal e é chamada no código main.

7. Edição do código da solução

O código desenvolvido como solução apresentava necessidade de correção para se adequar as normas AAPCS. Para isso, os registradores R4 a R10 tiveram seus estados salvos sempre que utilizados no código. Através da realização de um PUSH nos registradores e a após sua utilização, um POP.

8. Teste e depuração

Para executar os testes deste laboratório, foram utilizados dois casos de teste. O primeiro consistia na definição da Imagem 0, que possui os parâmetros: WIDTH=4 e HEIGHT=3. O segundo caso de teste consistia em utilizar os parâmetros: WIDTH1=160 e HEIGHT1=120 para então, assim como no primeiro caso, apresentar o histograma na forma de texto no terminal I/O.