



UFES – Universidade Federal da Fronteira Sul
Curso de Ciência da Computação
Disciplina – Organização de Computadores
Trabalho 01 – Programação Assembly

Instruções:

Trabalho em dupla

Data de entrega parcial: 05/11/2020 (obrigatória)

Data de entrega final: 19/11/2020

Entregar o arquivo .asm através do ambiente Moodle

Nome do arquivo de entrega: seunome_matricula.asm

Itens avaliados:

- Atendimento das instruções e especificação;
- Funcionalidade e corretude;
- Organização e indentação do código fonte;
- Menus, mensagens e interface;
- Documentação e comentários do código fonte.

Especificação:

Faça um programa que implementa a gerencia de uma lista encadeada ordenada com capacidade de armazenar números inteiros.

O programa deve ser implementado utilizando o conjunto de instruções do RISC V (RV32I) e deve ser executado no simulador RARS.

A gerencia da memória pode ser realizada via chamada ao sistema operacional (ecall 9) ou diretamente utilizando o registrador sp. Cada elemento ocupa 8 células de memória (4 células para armazenar o número inteiro e 4 células para armazenar o ponteiro para o próximo elemento da lista).

Os valores a serem inseridos na lista devem ser obtidos via teclado utilizando a ecall 1.

Exemplo: A figura abaixo mostra 4 elementos presentes na lista ordenada (-6, 3, 11 e 19), os endereços de memória onde cada um está armazenado e o ponteiro para a posição de memória do próximo elemento da lista. No exemplo, ptr_start representa o endereço de memória inicial da lista.

ptr_start = [200]

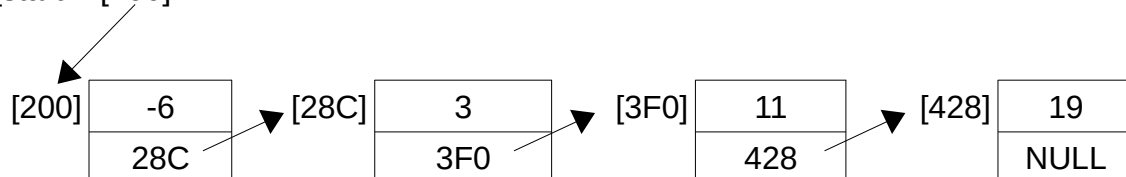


Figura 1: exemplo de lista encadeada ordenada com 4 elementos

O programa deve ter uma função *main*, a qual deve conter um menu com o acesso as seguintes funcionalidades:

- Inserir elemento na lista

- Recebe como parâmetros: posição de memória do primeiro elemento e o número inteiro a ser inserido;
- Retorna se operação foi realizada com sucesso ou não em a0; caso sucesso retorna a posição do elemento na lista em a0, caso de erro retorna -1; retorna em a1 o endereço inicial da lista;
- Remover elemento da lista por índice
 - Recebe como parâmetros: posição de memória do primeiro elemento e o índice do elemento a ser removido;
 - Retorna se operação foi realizada com sucesso ou não em a0; caso sucesso mostra o número retirado da lista em a1;
- Remover elemento da lista por valor (primeiro encontrado)
 - Recebe como parâmetros: posição de memória do primeiro elemento e o número a ser removido;
 - Indica se operação foi realizada com sucesso ou não; caso sucesso mostra o índice do valor retirado da lista;
- Listar todos os elementos da lista
 - Imprime na tela todos os elementos da lista;
 - Recebe como parâmetros: posição de memória do primeiro elemento;
- Sair do programa
 - Encerra a execução do programa mostrando a quantidade total de elementos inseridos e removidos

Segue o protótipo de cada uma das funções a serem implementadas pelo programa:

`int insere(int _lista[], int _elemento);`

- (a0) retorna -1 caso não tenha sido possível inserir na lista e o índice da posição inserida em caso de sucesso ;
- (a1) retorna o endereço de memória do início da lista
- a função deve inserir o elemento de forma ordenada na lista;

`int remove_indice(int _lista[], int indice);`

- (a0) retorna -1 caso não tenha sido possível retirar da lista;
- (a1) retorna o elemento retirado da posição em caso de sucesso;

`int remove_valor(int _lista[], int valor);`

- (a0) retorna -1 caso não tenha sido possível retirar da lista e o índice do elemento retirado caso contrário;
- deve retirar o primeiro elemento encontrado com o valor informado presente na lista;

`void imprime_lista(int _lista[]);`

- a função deve mostrar na tela todos os elementos presentes na lista;

Devem ser utilizadas as convenções de passagem e retorno de parâmetros do processador RISC V.

Podem ser criadas outras funções auxiliares caso julgar necessário.

O programa deve retornar ao menu principal depois de cada operação, encerrando-se somente quando o usuário solicitar (ocasião em que deve ser apresentada a quantidade total de elemetos inseridos na lista e a quantidade total de elementos removidos da lista);