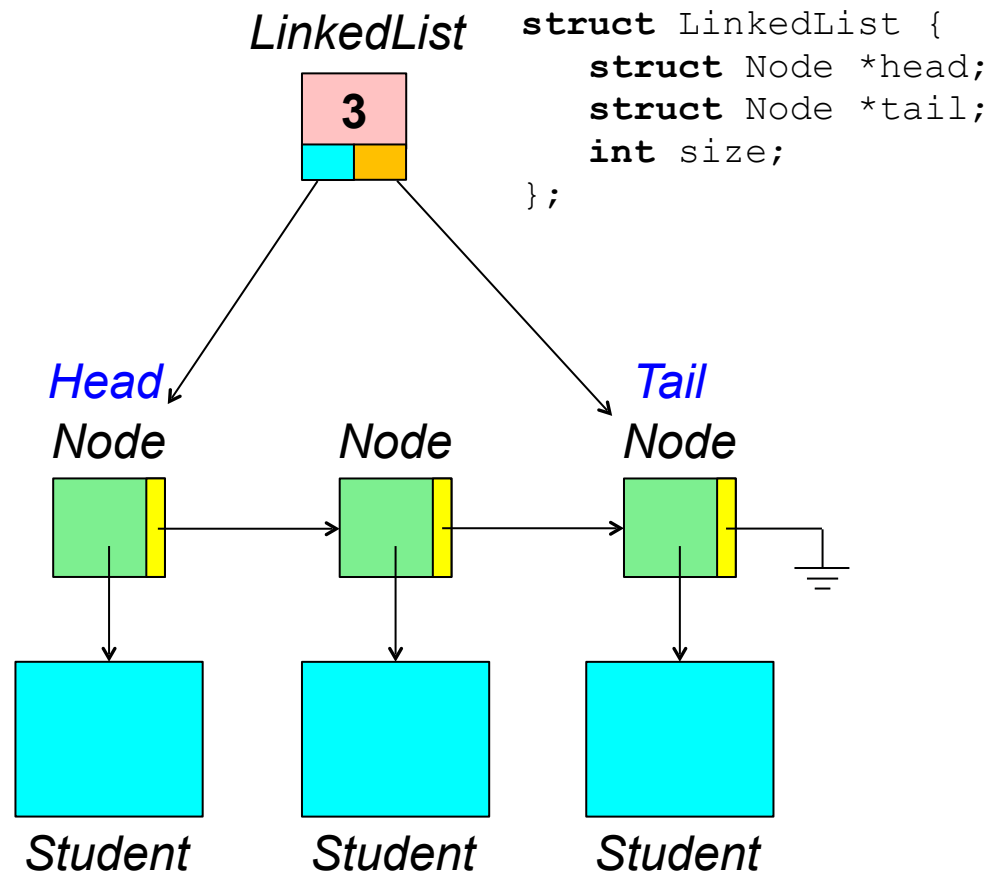


Trabalho

- Implementação em *assembly* de funções que operam sobre uma lista encadeada

```
struct Node {  
    void *data;  
    struct Node *next;  
};
```

```
struct Student {  
    char course[45];  
    unsigned int registration;  
    char name[130];  
  
    /* Birth date */  
    unsigned int year;  
    unsigned char month;  
    unsigned char day;  
  
    char hometown[20];  
};
```



Trabalho

- Moodle: LinkedList.c
 - Executar e entender as funções do programa

 - O programa implementa 5 funções em C que operam sobre listas encadeadas
 - *CreateList()*: Inicializa a lista
 - *CreateStudent()*: Instancia um estudante e preenche suas informações
 - *InsertStudent()*: Insere um estudante na lista (ordenada)
 - *PrintList()*: Imprime a lista do primeiro ao último estudante
 - *PrintListReverse()*: Imprime a lista do último ao primeiro estudante
-

Trabalho

- Implementar em *assembly* do MIPS a função *CreateStudent()* → *CreateStudent_asm()*
 - A função cria uma instância de estudante, preenche os campos e retorna um ponteiro para a instância preenchida
 - Na implementação, esta função **chama** as funções da biblioteca C:
 - *malloc()*: para alocar memória para o novo estudante
 - *strcpy()*: setar os campos *name*, *course* e *hometown*
 - **As funções da biblioteca C não devem ser implementadas. Devem ser apenas chamadas**
 - 3,0
-

Trabalho

- Implementar em *assembly* do MIPS a função *InsertStudent()* → *InsertStudent_asm()*
 - A função insere estudantes na lista de acordo com o número da matrícula
 - O estudante com a menor matrícula é o primeiro da lista (*head*)
 - Os demais são encadeado em ordem crescente
 - O estudante com a maior matrícula é o último da lista (*tail*)
 - Na implementação, esta função **chama** a função:
 - *malloc()*: para alocar memória para um novo nodo da lista
 - 4,0
-

Trabalho

- Implementar em *assembly* do MIPS a função *PrintListReverse()* → *PrintListReverse_asm()*
 - A função imprime os dados dos estudantes, do último (*tail*) ao primeiro (*head*)
 - Na implementação, esta função **chama** a função:
 - *printf()*: para imprimir uma *string*
 - A variável local *buffer*, declarada nesta função, deve ser alocada no *stack frame* da função *PrintListReverse_asm()*
 - 3,0

Trabalho

- Criar um projeto no MipsIt baseado no programa `LinkedList.c` para o desenvolvimento do trabalho
 - Substituir as funções *CreateStudent*, *InsertStudent* e *PrintListReverse* implementadas em C pelas respectivas implementações em *assembly*
 - Depois de substituídas as implementações em C pelas respectivas implementações em *assembly*, o resultado da execução deve ser idêntico
 - Dica
 - Substituir uma função por vez
 - A cada substituição, executar o programa e verificar se a saída continua correta
-

Trabalho

- **Atenção aos comentários do código *assembly***
 - Tomar como exemplo bubbleSort.asm (*moodle*)
 - Utilizar trechos do código C nos comentários quando for conveniente
 - **Códigos mal comentados valem no máximo 50%**
- **Atenção às convenções de uso dos registradores e da pilha (fundamental)**
- ***As funções/structs* já implementadas em C não devem ser alteradas em hipótese alguma (declaração/implementação)**

Trabalho

- Grupos de até 3 alunos
 - Evitar interação entre grupos com relação a código a fim de evitar que os trabalhos fiquem “parecidos demais”
 - **Não compartilhem código**
 - **IMPORTANTE**
 - **Não usem nenhum compilador para geração automatizada de código *assembly***
 - **Os códigos *assembly* devem ser gerados manualmente**
-

Trabalho

- Entrega: 10/12 (2,5 semanas)
 - Via *moodle*
 - *CreateStudent_asm.s*
 - *InsertStudent_asm.s*
 - *PrintListReverse_asm.s*
 - Arquivo texto descrevendo
 - As dificuldades enfrentadas durante o desenvolvimento
 - Quais funções funcionam e quais não
 - Colocar os nomes dos integrantes do grupo no cabeçalho dos arquivos
 - Apenas um integrante deve ser o responsável pela submissão
-