Relatório de Construção da Agenda de Compromissos: AppointmentAgenda

Luiz Felipe Diniz Costa 13782032

Rafael Jun Morita 10845040

Índice

1. Introdução

O presente relatório descreve o desenvolvimento do programa AppointmentAgenda, realizado como parte do trabalho da disciplina de Organização e Arquitetura de Computadores no Instituto de Ciências Matemáticas e de Computação (ICMC). O objetivo deste projeto foi criar uma ferramenta de gerenciamento de eventos em linguagem Assembly MIPS, proporcionando uma maneira eficiente e organizada de agendar compromissos.

2. Solução proposta

O AppointmentAgenda foi projetado visando tornar-se uma solução completa e intuitiva para a gestão de eventos. O objetivo principal foi oferecer aos usuários uma ferramenta completa e de fácil utilização. Dessa forma, a prioridade na construção deste programa foi garantir a ausência de conflitos na agenda, incorporando um sistema inteligente de detecção de sobreposições de horários, assegurando que os eventos agendados não entrem em conflito, mantendo assim a organização e a integridade do planejamento dos eventos.

Essa solução proporciona a flexibilidade necessária para ajustar e remover eventos conforme as demandas de uma agenda dinâmica, garantindo uma experiência eficiente e livre de complicações para os usuários, como será detalhado no próximo tópico.

2.1. Funcionalidades

Detalhes do Evento:
Cada evento registrado no AppointmentAgenda contém informações essenciais, como nome, data, hora de início e hora de término. Esses detalhes fornecem uma visão abrangente do cronograma de compromissos.

Resolução de Conflitos:

Implementamos um sistema automático de detecção de conflitos. Se um novo evento entra em conflito com um evento existente em termos de horário, o programa impede o registro desse novo evento. Isso garante que não haja sobreposições ou conflitos de agendamento.

Modificação de Eventos:
Os usuários têm a capacidade de editar e modificar eventos previamente registrados. Essa funcionalidade é essencial para fazer ajustes nos detalhes dos eventos, como alterar o nome, a data ou o horário, proporcionando flexibilidade ao gerenciamento da agenda.

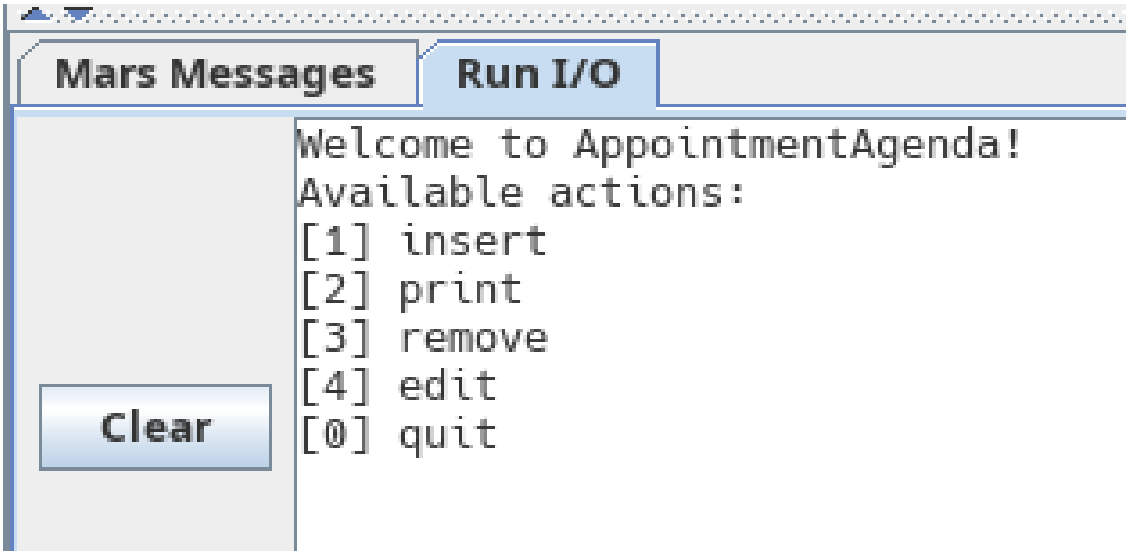
Remoção de Eventos:
O programa permite a remoção de eventos. Isso possibilita que eventos desatualizados ou cancelados sejam facilmente removidos da agenda, mantendo-a organizada e precisa.

Organização Cronológica:
A organização dos eventos é apresentada de forma cronológica, exibindo-os ordenadamente com base na data e hora de realização. Isso proporciona aos usuários uma visualização clara e sequencial de todos os compromissos agendados, facilitando o acompanhamento do cronograma de atividades.

Detecção de Inputs Inválidos:
Para garantir a precisão e integridade dos dados inseridos, o sistema é capaz de identificar inputs inválidos. Caso o usuário insira um dia inválido, horas fora do intervalo correto ou configure a hora de término antes da hora de início, a agenda prontamente emite um alerta indicando a entrada inválida. Além disso, inputs que contenham minutos excedendo 59, como por exemplo "10.60", são imediatamente identificados como erro. Essa funcionalidade de detecção contribui significativamente para prevenir inconsistências nos dados, garantindo que apenas informações precisas e pertinentes sejam registradas e mantendo a integridade dos dados armazenados na agenda.

2.2. Questões técnicas

Para elaborar o fluxo de trabalho da agenda, foi desenvolvido um menu composto por 5 opções destinadas aos usuários. Esta estrutura pode ser visualizada na imagem a seguir:



Cada opção desse menu representa uma das funções principais do programa, que são chamadas quando o usuário digita o número correspondente, internamente as funções principais podem usar tanto as outras principais como as secundárias no processo de execução da tarefa. Importante ressaltar que como estamos trabalhando com programação de baixo nível, tudo é feito explicitamente por meio de “jumps”.

3. Processamento da informação

3.1. Dados de entrada

Ao receber os dados requisitados, a etapa inicial consiste no armazenamento desses dados em variáveis auxiliares. Essa abordagem permite a realização de verificações essenciais antes do momento de inserção efetiva na memória, momento em que o contador de eventos é incrementado de fato.

Cada categoria de informação referente aos eventos é alocada em seu respectivo vetor, permitindo a associação entre os eventos através dos índices correspondentes. Essa correlação é estabelecida mediante a utilização da função secundária

"compareDay", a qual se encarrega de resolver possíveis sobreposições entre os eventos registrados. Para alcançar a ordenação dos eventos por dia, crucial para uma visualização otimizada na função de impressão, a função "compareDay" aciona a função "sortArray". Esse procedimento contribui significativamente para um gerenciamento mais eficiente e estruturado dos eventos, resultando em uma agenda organizada e de fácil compreensão. Isso será muito útil na implementação descrita no tópico 3.3.

3.2. Processamento

Para realizar a remoção de eventos, é necessário capturar de maneira precisa o evento que se deseja remover. Para alcançar isso, decidimos utilizar o índice do array como identificador único de cada evento. Dessa forma, cada evento é associado a um identificador baseado na sua posição no array. Para efetuar a remoção, é necessário deslocar todos os elementos à esquerda do evento que será removido no array. Após esse deslocamento, é feita a redução do contador de eventos, atualizando-o adequadamente.

No caso da função de edição de eventos, o processo foi um pouco mais complexo. Inicialmente, foi crucial capturar todos os dados do evento que se desejava editar, possibilitando ao usuário a opção de manter determinadas informações inalteradas durante a edição. Para realizar a edição, primeiramente acionamos a função de remoção para excluir o evento que será editado. Posteriormente, inserimos novamente o evento editado, proporcionando ao usuário a liberdade de manter ou alterar os dados conforme sua preferência. Essa ação de adicionar novamente o evento foi fundamental para reordenar todos os eventos, caso o usuário tenha alterado informações que impactassem a ordem correta dos eventos armazenados.

3.3. Dados de saída

Por esse motivo, na função print, há uma iteração por todos os elementos de cada array por meio de uma variável auxiliar. Enquanto imprime as informações dos compromissos, essa variável é incrementada. O processo continua até que o contador auxiliar alcance o mesmo valor do contador de eventos, momento em que a iteração é encerrada. Esse mecanismo assegura que todos os compromissos sejam exibidos corretamente e que a impressão seja concluída de forma precisa e abrangente.

Finalizando a descrição das funçoẽs principais, temos a função quit, que desempenha exclusivamente o papel de encerrar o programa.

4. Conclusão

A proposta do AppointmentAgenda visa oferecer aos usuários uma ferramenta completa e intuitiva para o gerenciamento de compromissos, demonstrando-se capaz de realizar o agendamento, modificação e remoção de eventos de forma eficiente. A ênfase na prevenção de conflitos por meio de um sistema inteligente de detecção de sobreposições de horários garantiu uma agenda organizada e livre de conflitos, proporcionando aos usuários uma experiência de agendamento tranquila e livre de complicações.

As funcionalidades implementadas, como a organização cronológica dos eventos, a detecção de inputs inválidos e os processos de remoção e edição, foram fundamentais para a precisão e integridade dos dados armazenados. A estruturação do programa com base em arrays e índices permitiu uma gestão eficaz dos eventos, garantindo a ordenação adequada e facilitando a visualização dos compromissos.

O desenvolvimento desse projeto foi um desafio significativo para os integrantes da equipe, sendo a primeira vez que trabalharam com uma linguagem de baixo nível como o Assembly MIPS. Lidar com uma problemática complexa, envolvendo considerável manipulação de memória, proporcionou não apenas aprendizado, mas também insights valiosos sobre o funcionamento interno dos sistemas computacionais.

5. Código Fonte

Acesse o projeto final, incluindo as instruções de uso e instalação, clicando neste link.