

Máquina de Café

Trabalho final de Organização de Computadores I

Juliana Miranda Bosio, Vinícios Rosa Buzzi

¹Departamento de Informática e Estatística – Universidade Federal de Santa Catarina
Caixa Postal 5094 – 88035-972 – Florianópolis – SC – Brazil

1. Introdução

A automação de processos tem se tornado uma tendência crescente em diversos setores, incluindo o de alimentos e bebidas. Neste contexto, a implementação de uma máquina de café automatizada representa uma aplicação prática e eficiente de conceitos de organização de computadores. Este relatório apresenta o desenvolvimento de uma máquina de café em linguagem assembly MIPS, utilizando o simulador MARS, com o intuito de simular o controle de hardware e software.

O sistema foi projetado para operar em ambientes comerciais, permitindo que os usuários selecionem diferentes tipos de café, tamanhos de copos e opções de adoçante. Além disso, a máquina é capaz de verificar a disponibilidade de ingredientes no estoque antes da preparação da bebida, garantindo que o usuário receba um produto de qualidade. A interação com o usuário é facilitada por meio de mensagens de confirmação e erro, que são exibidas em tempo real.

Este projeto não apenas visa a criação de uma máquina funcional, mas também a integração de conceitos fundamentais de organização de computadores, como manipulação de registradores, controle de fluxo, uso de pilha e chamadas de sistema (syscalls) para entrada e saída. Através deste trabalho, espera-se proporcionar uma compreensão mais profunda dos princípios de programação em assembly e das interações entre hardware e software.

2. Objetivo

O objetivo geral deste projeto é desenvolver uma máquina de café automatizada em linguagem assembly MIPS, utilizando o simulador MARS, com foco na simulação do controle de hardware e software.

2.1. Objetivos Específicos

Os objetivos específicos incluem:

- Implementar a lógica de seleção de diferentes tipos de café, tamanhos de copos e opções de adoçante.
- Verificar a disponibilidade de ingredientes no estoque antes da preparação da bebida.
- Exibir mensagens de confirmação ou erro em tempo real, proporcionando uma interação clara com o usuário.
- Utilizar chamadas de sistema (syscalls) para manipulação de entrada e saída, garantindo a comunicação eficaz entre o usuário e o sistema.
- Integrar conceitos de organização de computadores, como manipulação de registradores, controle de fluxo e uso de pilha, para garantir a eficiência e a funcionalidade do sistema.

2.2. Metodologia

A metodologia adotada para o desenvolvimento do projeto da máquina de café em assembly MIPS foi estruturada em etapas sequenciais, visando garantir uma abordagem organizada e eficiente. As etapas são descritas a seguir:

2.2.1. Definição do Problema e Requisitos

A primeira etapa consistiu em identificar o problema a ser resolvido e definir os requisitos do sistema. Isso envolveu a elaboração de uma lista de funcionalidades que a máquina de café deveria ter. Os principais requisitos incluem:

- Permitir que o usuário escolha entre diferentes tipos de bebidas, tamanhos de copos e opções de adoçantes.
- Verificar se os ingredientes necessários estão disponíveis antes de iniciar a preparação da bebida.
- Implementar uma lógica que simule a interação com o usuário, como a exibição de mensagens em um display e a captura de entradas por meio de um teclado.

2.2.2. Planejamento da Arquitetura do Sistema

Após a definição dos requisitos, foi planejada a estrutura do sistema. A arquitetura foi organizada em estados conforme figura abaixo.

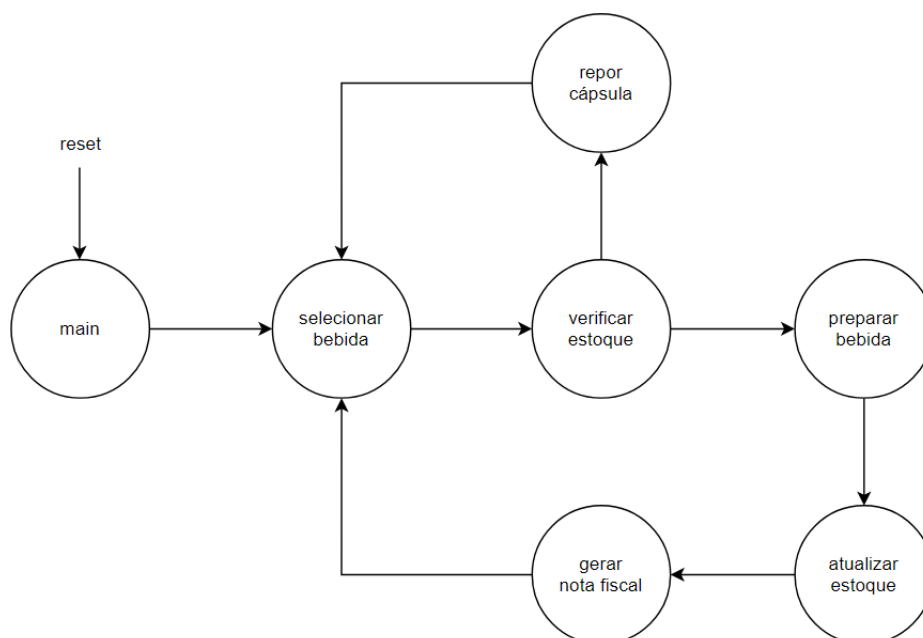


Figure 1. Diagrama de Estados Cafeteira

Cada funcionalidade seria implementada como uma sub-rotina em assembly. Os principais módulos planejados foram:

- Rotinas para exibir mensagens e capturar as escolhas do usuário.
- Módulos para verificar a validade das entradas e gerenciar possíveis erros.
- Funções para checar a disponibilidade dos ingredientes no estoque.
- Lógica para simular o preparo da bebida, incluindo o tempo necessário para cada etapa.

2.2.3. Prototipação do Código em Python

Para fins de visualização e planejamento do funcionamento da máquina de café, o código foi inicialmente prototipado em Python. Esta abordagem permitiu testar a lógica e a estruturação das operações, e também, criar uma visão geral do projeto comum entre os autores, antes da implementação em Assembly. A escolha do Python se deu pela sua simplicidade. Essa etapa garantiu que a transição para Assembly fosse mais eficiente com menor risco de falhas.

2.2.4. Implementação do Código em Assembly MIPS

Com a arquitetura definida, o próximo passo foi a implementação do código em linguagem assembly MIPS, utilizando o simulador MARS (MIPS Assembler and Runtime Simulator). Durante essa fase, foram seguidas algumas práticas importantes:

- Utilização de registradores para armazenar temporariamente dados, como as escolhas do usuário e os ingredientes disponíveis.
- Uso da pilha para salvar e restaurar informações durante as chamadas de funções, garantindo que o sistema mantenha o controle adequado do fluxo de execução.
- Estruturação do código de forma modular, permitindo que diferentes partes do programa fossem reutilizadas e facilitando a manutenção.

2.2.5. Simulação e Testes

Após a implementação, a simulação foi realizada no MARS, que possibilitou:

- Testar o funcionamento do sistema em diferentes cenários, como entradas válidas e inválidas, para garantir que a máquina respondesse corretamente.
- Identificar e corrigir erros de lógica, observando o comportamento do sistema durante a execução.
- Monitorar em tempo real a sequência de instruções e os valores armazenados nos registradores, permitindo uma análise detalhada do funcionamento do código.
- Verificar se as mensagens de confirmação e erro eram exibidas corretamente, assegurando uma boa experiência para o usuário.

2.3. Importância da Simulação

A simulação foi uma etapa crucial no desenvolvimento do projeto da máquina de café, pois permitiu testar e validar o programa de forma prática. Utilizando o MARS, foi possível aplicar diversos conceitos aprendidos na disciplina, como o uso de registradores para armazenamento temporário de dados, o controle de fluxo com instruções condicionais e de salto, e o gerenciamento de chamadas de procedimentos através da pilha.

Além disso, a simulação facilitou o tratamento de entradas e saídas por meio de chamadas de sistema (syscalls), reforçando a compreensão da arquitetura MIPS e sua execução em um ambiente simulado. Essa experiência prática não apenas consolidou o conhecimento teórico, mas também preparou os alunos para desafios futuros na área de programação e sistemas embarcados.

3. Estrutura do Código

O código é organizado em várias seções, cada uma responsável por uma funcionalidade específica. As principais seções incluem:

- **Dados:** Definição de variáveis e mensagens utilizadas no sistema.
- **Texto:** Implementação das rotinas de controle, incluindo seleção de bebida, tamanho, adoçante, verificação de estoque, preparação da bebida e geração do comprovante.

4. Chamadas de Sistema Utilizadas

O código utiliza diversas chamadas de sistema (syscalls) para interagir com o usuário e controlar a máquina. As principais syscalls utilizadas incluem:

- **Syscall 4:** Exibe mensagens na tela, como prompts e confirmações.
- **Syscall 1:** Lê a entrada do usuário, permitindo a seleção de opções.
- **Syscall 13:** Abre ou cria um arquivo para gerar o comprovante fiscal.
- **Syscall 15:** Escreve dados no arquivo, registrando as informações da compra.
- **Syscall 16:** Fecha o arquivo após a escrita.
- **Syscall 30:** Utilizada para implementar um timer, controlando o tempo de preparo das bebidas.

5. Conceitos Utilizados

A implementação da máquina de café abrange diversos conceitos fundamentais de organização de computadores:

- **Manipulação de Registradores:** O código utiliza registradores para armazenar temporariamente dados, como a quantidade de ingredientes e as seleções do usuário.
- **Controle de Fluxo:** Instruções condicionais e de salto são utilizadas para gerenciar a lógica do programa, permitindo que o sistema responda a diferentes entradas do usuário.
- **Uso de Pilha:** A pilha é utilizada para salvar e restaurar o endereço de retorno durante chamadas de procedimentos, garantindo que o fluxo do programa seja mantido.
- **Tratamento de Entradas e Saídas:** O sistema utiliza syscalls para interagir com o usuário e manipular arquivos, demonstrando a importância do gerenciamento de I/O em sistemas computacionais.
- **Estrutura Modular:** O código é organizado em sub-rotinas, cada uma responsável por uma funcionalidade específica, promovendo a reutilização e a clareza do código.

6. Fluxo de Operação

O fluxo de operação da máquina de café pode ser descrito nas seguintes etapas:

1. O usuário inicia o sistema e é apresentado ao menu de seleção de bebidas.
2. O usuário escolhe o tipo de bebida desejada.
3. O sistema solicita a seleção do tamanho da bebida.
4. O usuário opta por adicionar açúcar ou não.
5. O sistema verifica a disponibilidade dos ingredientes necessários para a preparação da bebida.
6. Se os ingredientes estiverem disponíveis, o sistema inicia a preparação e informa o usuário.
7. Após a preparação, um comprovante fiscal é gerado e salvo em um arquivo .txt.
8. O sistema retorna ao menu inicial, permitindo que o usuário faça um novo pedido.

7. Limitações do Sistema

Embora o sistema de pedidos de café implementado em MIPS seja funcional, ele apresenta algumas limitações que podem impactar a experiência do usuário e a eficiência do sistema. As principais limitações incluem:

- **Interação com o Teclado Digital:** O sistema requer que o usuário clique duas vezes em uma tecla do teclado digital para deselegionar uma opção antes de selecionar um novo item. Essa interação pode ser confusa e não intuitiva, levando a erros na seleção de opções.
- **Limitação de Instruções por Segundo:** O sistema é projetado para operar com um limite de menos de 30 instruções por segundo. Essa restrição pode afetar a fluidez da interface do usuário, especialmente em situações onde múltiplas entradas são processadas rapidamente.
- **Caminho do Arquivo:** O caminho para o arquivo de saída (nota fiscal) deve ser atualizado manualmente no código para corresponder ao sistema de arquivos da máquina em que o programa está sendo executado. Isso pode causar dificuldades na portabilidade do código entre diferentes ambientes de desenvolvimento.
- **Dependência de Recursos de Hardware Específicos:** O código depende de um hardware específico para a interação com displays de sete segmentos e teclado digital. Isso limita a capacidade de executar o código em diferentes plataformas ou emuladores que não suportam esses dispositivos.
- **Gerenciamento de Estoque:** O sistema não possui um mecanismo automatizado para gerenciar o estoque de ingredientes. A reposição de ingredientes é feita manualmente, o que pode levar a interrupções no serviço se não for monitorado adequadamente.
- **Interface do Usuário:** A interface do usuário é baseada em texto e não oferece uma experiência visual interativa. Isso pode ser um obstáculo para usuários que preferem interfaces gráficas mais intuitivas.
- **Limitações de Escalabilidade:** O sistema é projetado para um número fixo de opções de bebidas e tamanhos. A adição de novas opções exigiria modificações no código, o que pode ser um desafio em um ambiente de produção.

Essas limitações devem ser consideradas ao avaliar a eficácia e a usabilidade do sistema, e melhorias podem ser implementadas para superar esses desafios em versões futuras.

8. Conclusão

A implementação da máquina de café em assembly MIPS não apenas consolidou os conhecimentos adquiridos na disciplina de Organização de Computadores, mas também proporcionou uma experiência prática na programação de sistemas embarcados. O projeto demonstrou a importância da lógica de programação, controle de hardware e software, e a interação com o usuário em um ambiente simulado. Através deste trabalho, foi possível explorar conceitos fundamentais de arquitetura de computadores e suas aplicações práticas.

9. Repositório da Atividade

Confira os códigos implementados acessando o *QRCode* abaixo ou pelo link:

[https://github.com/buzziologia/UFSC/tree/main/
OrganizacaoDeComputadores/TrabalhoFinal](https://github.com/buzziologia/UFSC/tree/main/OrganizacaoDeComputadores/TrabalhoFinal)



Figure 2. Repositório Github

Anexo A: Caso de Teste - Negar Café Grande com Estoque Limitado

Descrição: Verificar se o sistema nega a opção de café grande quando há apenas uma cápsula disponível, mas permite a seleção de café pequeno.

Configuração Inicial:

- $s_0 = 1$ (estoque de café)
- $s_1 = 20$ (estoque de leite)
- $s_2 = 20$ (estoque de chocolate)
- $s_3 = 20$ (estoque de açúcar)

Sequência de Pedidos:

1. Iniciar a máquina de café.
2. Selecionar a opção de café grande.
3. Confirmar o pedido.
4. Observar a mensagem de erro exibida.
5. Selecionar a opção de café pequeno.
6. Confirmar o pedido.
7. Observar a mensagem de confirmação.

Resultado Esperado:

- O sistema deve negar o pedido de café grande e exibir uma mensagem informando que não há cápsulas suficientes.
- O sistema deve permitir o pedido de café pequeno e exibir uma mensagem de confirmação.

Resultados:

Anexo B: Caso de Teste - Negar Reposição de Estoque

Descrição: Verificar se o sistema nega a reposição de estoque para 20 cápsulas quando ainda há uma cápsula disponível.

Configuração Inicial:

- \$s0 = 1 (estoque de café)
- \$s1 = 20 (estoque de leite)
- \$s2 = 20 (estoque de chocolate)
- \$s3 = 20 (estoque de açúcar)

Passos:

1. Iniciar a máquina de café.
2. Tentar pedir um café grande (sequência proposta = 2, 2, 2).
3. observar a mensagem no prompt informando insuficiência de cápsulas.
4. informar a tecla A como senha de reposição.
5. observar mensagem exibida.

Resultado Esperado:

- O sistema deve exibir uma mensagem informando que a reposição falhou e que voltou para o menu principal, uma vez que ainda há cápsula disponível no estoque.

Resultados:

Anexo C: Caso de Teste - Sequências de Pedidos e Atualização de Estoque

Descrição: Demonstrar várias sequências de pedidos para verificar a correta atualização e verificação do estoque.

Configuração Inicial:

- \$s0 = 20 (estoque de café)
- \$s1 = 20 (estoque de leite)
- \$s2 = 20 (estoque de chocolate)
- \$s3 = 20 (estoque de açúcar)

Passos:

1. Iniciar a máquina de café.
2. Selecionar e confirmar o pedido de 1 Mocaccino pequeno com açúcar (sequência do pedido 3, 1, 2)
3. Selecionar e confirmar o pedido de 1 Café puro grande com açúcar (sequência do pedido 1, 2, 2)
4. Selecionar e confirmar o pedido de 1 Café com leite pequeno sem açúcar (sequência do pedido 2, 1, 1)
5. Verificar o estoque (registradores).

Resultado Esperado:

- O sistema deve atualizar o estoque corretamente após cada pedido.

Valores Esperados ao Final da Simulação:

- \$s0 = 16 (estoque total de cápsulas de café após 3 pedidos)
- \$s1 = 18 (1 cápsula de leite consumida para o café pequeno)
- \$s2 = 19 (1 cápsula de chocolate consumida para o café médio)
- \$s3 = 17 (1 cápsula de açúcar consumida para o café grande)

Resultados:

Anexo D: Caso de Teste - Opções de Desistir do Pedido

Descrição: Verificar se o sistema permite ao usuário desistir do pedido antes da confirmação.

Configuração Inicial:

- \$s0 = 20 (estoque de café)
- \$s1 = 20 (estoque de leite)
- \$s2 = 20 (estoque de chocolate)
- \$s3 = 20 (estoque de açúcar)

Passos:

1. Iniciar a máquina de café.
2. Selecionar a opção de café pequeno.
3. Antes de confirmar, escolher a opção de desistir do pedido.
4. Observar a mensagem exibida.

Resultado Esperado:

- O sistema deve permitir que o usuário desista do pedido e exibir uma mensagem confirmando a desistência.
- Verificar se o sistema retorna ao menu principal após a desistência.
- Confirmar se o estoque permanece inalterado após a desistência do pedido.

Resultados:

Anexo E: Caso de Teste - Verificação de Entradas Válidas

Descrição: Verificar se o sistema aceita apenas entradas válidas do usuário.

Configuração Inicial:

- \$s0 = 20 (estoque de café)
- \$s1 = 20 (estoque de leite)
- \$s2 = 20 (estoque de chocolate)
- \$s3 = 20 (estoque de açúcar)

Passos:

1. Iniciar a máquina de café.
2. Tentar inserir uma opção inválida (por exemplo, um número fora do intervalo de opções disponíveis).
3. Observar a mensagem exibida.
4. Inserir uma opção válida.
5. Confirmar o pedido e observar a mensagem de confirmação.

Resultado Esperado:

- O sistema deve exibir uma mensagem de erro ao receber uma entrada inválida.
- O sistema deve aceitar a entrada válida e exibir uma mensagem de confirmação.
- Verificar se o sistema permite a repetição da entrada após uma tentativa inválida.
- Confirmar se o estoque permanece inalterado após a tentativa de entrada inválida.

Resultados:

Anexo F: Caso de Teste - Reposição Correta do Estoque

Descrição: Verificar se o sistema realiza a reposição correta do estoque após a adição de novos itens.

Configuração Inicial:

- \$s0 = 1 (estoque de café)
- \$s1 = 20 (estoque de leite)
- \$s2 = 0 (estoque de chocolate)
- \$s3 = 20 (estoque de açúcar)

Passos:

1. Iniciar a máquina de café.
2. Registrar os níveis atuais de estoque.
3. Selecionar e confirmar o pedido de 1 Mocaccino pequeno sem açúcar (sequência do pedido 3, 1, 1).
4. Observar a mensagem de insuficiência de cápsulas.
5. Informar a senha A para reportar o estoque.
6. Selecionar e confirmar o pedido de 1 Mocaccino pequeno sem açúcar (sequência do pedido 3, 1, 1).
7. verificar confirmação do pedido.
8. Selecionar e confirmar o pedido de 1 Mocaccino pequeno sem açúcar (sequência do pedido 3, 1, 1).
9. Observar a mensagem de insuficiência de cápsulas (dessa vez café está em falta).
10. Informar a senha A para reportar o estoque.
11. Observar os registradores de cápsulas.

Resultado Esperado:

- \$s0 = 20 (estoque de café)
- \$s1 = 19 (estoque de leite)
- \$s2 = 19 (estoque de chocolate)
- \$s3 = 20 (estoque de açúcar)

Resultados:
