Relatório do 1º trabalho prático de Organização e Arquitetura de Computadores

Integrantes do grupo:
Cauê Paiva Lira - 14675416
Luiz Felipe Diniz Costa - 13782032
Pedro Henrique Ferreira Silva - 14677526
Pedro Louro Fernandes - 13672446

Este relatório tem como objetivo descrever o processo de desenvolvimento do jogo, ao expor os desafios que surgiram ao decorrer do projeto, como os solucionamos e o aprendizado necessário para alcançar essas soluções.

Desafios enfrentados:

- Gerar um número aleatório entre 1 e 100;
- Armazenar e imprimir o número de tentativas do usuário;
- Encontrar uma forma de repetir a pergunta para o usuário quando ele erra o número;
- Armazenar e imprimir as tentativas do usuário em ordem;
- Dificuldade geral com a formatação de um código em Assembly.

Como nós solucionamos esses desafios:

Para gerar um número aleatório, nós geramos uma seed pseudo-aleatória usando a instrução syscall e depois aplicamos uma fórmula de algoritmo congruencial linear (ACL) para aumentar a aleatoriedade do valor gerado multiplicando por um número A e somando com um valor C, e depois calculando o resto da divisão por 100 e somando 1 para que o valor esteja entre 1 e 100. Criamos uma função para gerar o número e essa função retorna o número gerado em um registrador.

Para armazenar o número de tentativas do usuário no registrador s4, percorremos a lista de valores digitados pelo usuário depois que ele acerta o número e incrementamos uma unidade no registrador a cada valor da lista.

Para repetir a pergunta para o usuário quando ele erra, usamos as instruções j e blt para fazer um jump para outra parte do código se o valor digitado for maior ou menor que o número gerado, e quando o valor é igual, fazemos um jump para a parte que finaliza o jogo (victory).

Para armazenar as tentativas do usuário, usamos uma lista encadeada, em que cada nó contém o valor digitado e um ponteiro para o próximo nó da lista. Para armazenar um novo valor, usamos a instrução sw para salvar o valor para no endereço alocado na lista (sw t1, 0(a0)). E para imprimir os valores, carregamos o primeiro valor para o registrador t2 e se o nó não for null, imprime o inteiro e uma quebra de linha, move o t2 para o endereço do próximo nó e depois dá um jump incondicional para o início do loop.

Já a formatação foi um desafio por conta de seu formato diferente do comum em relação a outras linguagens de programação como Python e C. E diferentes editores como o RARS, VSCode e a interface do GitHub tem indentações um pouco diferentes, então tivemos que levar isso em consideração.

Conclusão

De modo geral, para este trabalho foi necessário pesquisar e aprender mais a fundo sobre o funcionamento da linguagem Assembly bem como sua manipulação de mais baixo nível dos recursos do computador. Um desses recursos é o manejo direto de registradores e elementos de memória, até então pouco utilizados a esse nível em outras linguagens.