Universidade Federal do Pará
Instituto de Ciências Exatas e Naturais
Faculdade de Ciência da Computação

DOCUMENTAÇÃO DO JOGO

Laboratório de Sistemas Operacionais

ALUNO: ARTHUR FEITOSA DA CONCEIÇÃO

HEITOR LIMA E SILVA

KALEO NABOR PIMENTEL DA CUNHA

YHANN MATHEUS DE DIO MIRANDA MENDES

BELÉM - PA
2025.

1. JOGO

"A Torre" é um jogo de aventura em texto onde o jogador controla Elric, um personagem que explora uma torre misteriosa em busca de seu filho perdido. O jogo apresenta diversas cenas narrativas e combates contra inimigos.

2. IMPLEMENTAÇÃO DAS THREADS

Thread dedicada exclusivamente para controle de áudio:

Semáforo sem_musica_mudanca: Sinaliza quando uma nova música deve ser tocada

Semáforo sem_musica_controle: Garante que apenas uma operação musical ocorra por vez

```c
// Thread que gerencia a música em background
void* thread_gerenciador_musica(void* arg) {
    while (thread_ativa) {
        sem_wait(&sem_musica_mudanca); // Espera sinal para trocar música

        if (!thread_ativa) break;

        sem_wait(&sem_musica_controle); // Semáforo para controle crítico

        if (musica_solicitada != musica_atual_tocando) {
            // Para música atual
            if (Mix_PlayingMusic()) {
                Mix_HaltMusic();
                usleep(50000); // 50ms de pausa
            }

            // Toca nova música se válida
            if (musica_solicitada != MUSICA_NENHUMA && musicas[musica_solicitada]) {
                if (Mix_PlayMusic(musicas[musica_solicitada], -1) == 0) {
                    musica_atual_tocando = musica_solicitada;
                } else {
                    musica_atual_tocando = MUSICA_NENHUMA;
                }
            } else {
                musica_atual_tocando = MUSICA_NENHUMA;
            }
        }

        sem_post(&sem_musica_controle);
    }
    return NULL;
}
```

3. SISTEMA DE ENTRADA ASSÍNCRONA:

Thread utilizada para captura de input:

Semáforo sem_input_ready: Indica que novo input está disponível

Semáforo sem_input_request: Solicita processamento do input

```
// Thread de input
while (jogo_ativo) {
    captura_input(); // Não-bloqueante
    if (novo_input) {
        sem_post(&sem_input_ready); // Avisa que tem input
        sem_wait(&sem_input_request); // Espera processamento
    }
}

// Thread principal
sem_wait(&sem_input_ready); // Espera input disponível
processa_input(); // Executa ação do jogador
sem_post(&sem_input_request); // Libera para próximo input
```

4. CONTROLE DE CONCORRÊNCIA

- Seções críticas protegidas por semáforos
- Sincronização coordenada entre threads principais e auxiliares
- Protocolo de finalização segura para todas as threads

```
// Acesso seguro a dados compartilhados
sem_wait(&sem_dados_jogador); // Entra na seção crítica
jogador->vida -= dano; // Modificação segura
sem_post(&sem_dados_jogador); // Sai da seção crítica

// Finalização ordenada
void finalizar_jogo() {
    jogo_ativo = false; // Sinal para todas as threads
    sem_post(&sem_musica_mudanca); // Libera threads bloqueadas
    sem_post(&sem_input_ready);
    // Aguarda todas as threads terminarem
    pthread_join(thread_musica, NULL);
    pthread_join(thread_input, NULL);
}
```