```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

typedef struct {
    int id;
    char descricao[100];
} Processo;

typedef struct no {
    Processo info;
    struct no* prox;
} No;

typedef struct {
    No* ini;
    No* fim;
} Fila;

Fila* cria_fila();
void incluir_processo(Fila* f, Processo p);
void retirar_processo(Fila* f);
void imprimir_fila(Fila* f);
void menu();

int main() {
    Fila* fila_de_processos = cria_fila();
    Processo p;
    int opcao;
    int proximo_id = 1;

    do {
        menu();
        scanf("%d", &opcao);

        switch (opcao) {
            case 1:
                p.id = proximo_id++;
                printf("Digite a descricao do processo: ");
                while (getchar() != '\n');
                fgets(p.descricao, 100, stdin);
                p.descricao[strcspn(p.descricao, "\n")] = 0;

                incluir_processo(fila_de_processos, p);
                printf("Processo #%d (%s) incluido na fila.\n\n", p.id, p.descricao);
                break;
            case 2:
                retirar_processo(fila_de_processos);
                break;
```

```
            case 3:
                imprimir_fila(fila_de_processos);
                break;
            case 0:
                printf("Encerrando o sistema...\n");
                break;
            default:
                printf("Opcao invalida!\n\n");
        }
    } while (opcao != 0);

    free(fila_de_processos);
    return 0;
}

void menu() {
    printf("--- Sistema de Gerenciamento de Processos ---\n");
    printf("1. Incluir novo processo na fila\n");
    printf("2. Retirar processo (com maior tempo de espera)\n");
    printf("3. Imprimir fila de processos\n");
    printf("0. Sair\n");
    printf("Escolha uma opcao: ");
}

Fila* cria_fila() {
    Fila* f = (Fila*)malloc(sizeof(Fila));
    f->ini = NULL;
    f->fim = NULL;
    return f;
}

void incluir_processo(Fila* f, Processo p) {
    No* novo = (No*)malloc(sizeof(No));
    novo->info = p;
    novo->prox = NULL;
    if (f->fim != NULL) {
        f->fim->prox = novo;
    } else {
        f->ini = novo;
    }
    f->fim = novo;
}

void retirar_processo(Fila* f) {
    if (f->ini == NULL) {
        printf("\nFila de processos esta vazia!\n\n");
        return;
    }
    No* aux = f->ini;
```

```
    Processo p_retirado = aux->info;

    f->ini = aux->prox;
    if (f->ini == NULL) {
        f->fim = NULL;
    }
    free(aux);

    printf("\nProcesso #%d (%s) foi retirado para execucao.\n\n", p_retirado.id,
p_retirado.descricao);
}

void imprimir_fila(Fila* f) {
    No* aux = f->ini;
    printf("\n--- FILA DE PROCESSOS ATUAL ---\n");
    if (aux == NULL) {
        printf("Nenhum processo na fila.\n");
    }
    while (aux != NULL) {
        printf("[ID: %d, Desc: %s] -> ", aux->info.id, aux->info.descricao);
        aux = aux->prox;
    }
    printf("FIM\n\n");
}
```