Trabalho 1 - Sistemas Operacionais

Guilherme Sousa Lopes - 535869
Lucas Rodrigues Aragão - 538390

Sumário

- Ideia da fila
- Pontos críticos
- Detalhes de implementação

FILA

Fila - Ideia

- Somente uma fila
- Implementada como um vetor de structs Cliente, de maneira “circular”
- FIFO.

Fila - Funções

A fila possui as funções de

- Inicializar
- Checar se está vazia
- Checar se está cheia
- Inserir
- Remover

Fila - Implementação

```c
typedef struct cliente {
    int pid;
    int hora_chegada;
    int prioridade;
    int tempo_atendimento;
} cliente;

typedef struct {
    cliente fila[101];
    int inicio;
    int fim;
} FilaCliente;

void inicializarFila(FilaCliente* f);
int filaVazia(FilaCliente* f);
int filaCheia(FilaCliente* f);
void inserir(FilaCliente* f, cliente c);
cliente remover(FilaCliente* f);
```

Pontos críticos do código

Controle de threads

- Semáforos
- Função Waitpid
- Usleep

Semáforos

- Sem_atend e Sem_block

- Sem_fila
  - Controlar o acesso à fila
  - Criado junto aos outros semáforos,a thread recepção
  - Usado nas ações de remoção e de inserção de elementos na fila

Waitpid

- No momento da criação de clientes era necessário esperar ele dormir para continuar a thread recepção e como não podemos alterar o cliente.c essa foi a ideia que surgiu.

```
[...]
waitpid(pid_cliente, &status, WUNTRACED);


[...]
waitpid(pid_cliente, &status, 0);
```

Usleep

- Usados em alguns laços while, apenas para aguardar em determinado ponto, como em

```
while(filaVazia(fila)){
        usleep(1); }
```

- Complicações...

```
void* atendente(void* args){
    usleep(1000);
[...] }
```

Detalhes de implementação

Detalhes de Implementação

- A criação de clientes foi feita usando fork
- O analista sempre é chamado a cada 10 clientes atendidos.
- O pid do analista é armazenado em um arquivo temporário criado no momento em que o analista é executado.
- Para medir os tempos no programa foi usada a função clock_gettime() da biblioteca time.h.
- Variável booleana pode_parar

Fim