Relatório Intermédio do Trabalho Prático

Urgências @ DEI

Sistemas Operativos

Ano Letivo 2025/2026 - PL7

Diogo Lemos nº 2020219666

UNIVERSIDADE D COIMBRA
FACULDADE DE CIÊNCIAS E TECNOLOGIA

Mecanismos de Sincronização

1. **Sincronização na Triage Queue**
    - Uso de Mutex para garantir que apenas uma thread manipula a queue de cada vez
    - Uso de Condition Variables para espera eficiente das threads por nova informação e para certificar que a queue não se encontra cheia.
2. **Sincronização na Triage Thread Pool**
    - Para gerir dinamicamente o número de Threads através de TRIAGE=X e acordar e adormecer threads quando necessário, é usado um mutex próprio da triagem para proteger o array de threads.
3. **Sincronização do Ficheiro Log MMF**
    - Existe um mutex global do log para garantir que não há perda de informação entre a escrita de múltiplas threads no ficheiro de log.
4. **Sincronização na criação dinâmica de Doctors**
    - A substituição de doctors após o fim de turno é possível através do uso de dois sinais: SIGALRM para detectar o fim do turno e SIGTERM para acabar o turno.
    - Para ocorrer a substituição do doctor terminado, é usado o handler SIGCHLD e há recolha não bloqueante do status dos processos terminados com waitpid(-1, WNOHANG).
5. **Sincronização da SHM - Estatísticas**
    - Para garantir que não há corrupção na escrita de dados na Shared Memory dos diversos tempos para cálculo das estatísticas, é necessária a existência de mutex.

6. **Sincronização do Named Pipe**
    - A função read() por si só é bloqueante, logo cada linha chega de forma consistente para a Admission enviar para a Triage Queue através da sua própria sincronização.
7. **Shutdown controlado**
    - Após SIGINT, temos keep_running = 0 para terminar o loop principal de admission, uma variável para indicar o término das threads, uma variável de condição para acordar as threads adormecidas na queue de triagem, join() das threads, SIGTERM para todos os doctors, wait() para recolher o estado dos processos filhos e a libertação dos recursos IPC de forma inversa à criação.

Decisões importantes

1. Escolhida fila circular para eficiência O(1) e uso fixo de memória
2. Uso de condition variables para eficiência energética e CPU
3. Uso de MMF para performance sob alta concorrência
4. Receção de prioridades negativas na MSQ, para que o valor -1 corresponda à maior prioridade.

Diagrama