Lista de Requisitos e Walkthrough Modular - Simulador de Núcleo de SO

Lista de Requisitos para Cada Módulo

1. Kernel (src/kernel/)

- Controlar a entrada de eventos através das funções sysCall() e interruptControl().
- Ativar as threads responsáveis pelo tratamento de cada evento.
- Manter o controle dos estados dos processos via BCP.
- Interagir com o Scheduler para escalar o próximo processo a ser executado.
- Interface pública: sysCall(int eventCode, int pid); interruptControl(int eventCode, int pid).

2. Gerenciador de Processos - BCP (src/kernel/)

- Estrutura BCP com PID, estado (Pronto, Executando, Bloqueado), prioridade, tempo de execução restante, etc.
- Funções para criar, bloquear, desbloquear e destruir processos.
- Interface pública: bcp_createProcess(int pid); bcp_blockProcess(int pid); bcp_unblockProcess(int pid); bcp_destroyProcess(int pid).

3. Scheduler (src/kernel/)

- Implementar um algoritmo de escalonamento (definir com o professor antes).
- Manter as filas de processos prontos.
- Escolher o próximo processo a ser executado quando o atual termina ou é bloqueado.
- Interface pública: scheduler_nextProcess(); scheduler_addProcess(int pid).

4. Gerenciador de Disco (src/disk/)

- Receber requisições de leitura e escrita (DISK_REQUEST).
- Simular tempo de operação (sleep ou relógio virtual).
- Chamar interruptControl(DISK_FINISH, pid) ao término da operação.
- Interface pública: handleDiskRequest(void* arg).

5. Gerenciador de Impressão (src/printer/)

Lista de Requisitos e Walkthrough Modular - Simulador de Núcleo de SO

- Receber requisições de impressão (PRINT_REQUEST).
- Simular tempo de impressão.
- Chamar interruptControl(PRINT_FINISH, pid) ao término da operação.
- Interface pública: handlePrintRequest(void* arg).

6. Gerenciador de Memória (src/memory/)
- Receber requisições de carregamento de memória (MEM_LOAD_REQ).
- Simular tempo de carregamento.
- Chamar interruptControl(MEM_LOAD_FINISH, pid) ao término da operação.
- Interface pública: handleMemLoadRequest(void* arg).

7. Gerenciador de Semáforos (src/semaphore/)
- Implementar funções P(s) e V(s) para bloquear e liberar processos.
- Controlar os semáforos disponíveis e filas de processos bloqueados.
- Interface pública: semaphoreP(int semId, int pid); semaphoreV(int semId, int pid).

8. Interpretador de Programas Sintéticos (src/synthetic_program/)
- Ler arquivos de programas sintéticos.
- Interpretar comandos e gerar eventos sysCall ou interruptControl.
- Controlar o avanço do tempo virtual.
- Interface pública: interpreter_loadProgram(const char* filename); interpreter_run().

9. Relógio Virtual (src/synthetic_program/)
- Controlar o tempo de execução dos processos (exec t).
- Informar ao interpretador e ao Kernel o momento de finalização de execuções.
- Interface pública: clock_advance(int timeUnits); clock_getTime().

Walkthrough Genérico por Módulo

Lista de Requisitos e Walkthrough Modular - Simulador de Núcleo de SO

Kernel:

1. Inicializar o sistema (initKernel).
2. Receber eventos de sysCall() e interruptControl().
3. Acionar a thread (ou função) que trata cada evento (ex: handleDiskRequest).
4. Atualizar estado dos processos no BCP.
5. Chamar o Scheduler após a conclusão de um evento.

Gerenciador de Processos (BCP):

1. Criar processos com seus atributos (PID, prioridade, estado inicial).
2. Gerenciar estados: pronto, executando, bloqueado.
3. Bloquear processos que pedem E/S ou aguardam semáforos.
4. Desbloquear processos ao fim das operações.

Scheduler:

1. Manter filas de processos prontos, por prioridade ou RR (a definir).
2. Após uma interrupção ou término, decidir quem será o próximo a executar.
3. Fornecer PID do próximo processo ao Kernel.

Gerenciador de Disco:

1. Ao receber DISK_REQUEST, bloquear o processo solicitante.
2. Simular o tempo de operação (sleep ou clock).
3. Ao final, enviar DISK_FINISH ao Kernel.

Gerenciador de Impressão:

1. Ao receber PRINT_REQUEST, bloquear o processo solicitante.
2. Simular o tempo de impressão.
3. Ao final, enviar PRINT_FINISH ao Kernel.

Gerenciador de Memória:

1. Ao receber MEM_LOAD_REQ, verificar espaço simulado na memória.

Lista de Requisitos e Walkthrough Modular - Simulador de Núcleo de SO

2. Simular o tempo de carregamento.

3. Ao final, enviar MEM_LOAD_FINISH ao Kernel.

Gerenciador de Semáforos:

1. Ao receber P(s), verificar se o recurso está livre.

2. Se não estiver, bloquear o processo.

3. Ao receber V(s), liberar o semáforo e acordar o próximo processo na fila.

Interpretador de Programas Sintéticos:

1. Ler o arquivo .txt que contém os comandos do processo.

2. Interpretar cada linha e disparar sysCall() ou interruptControl() de acordo com o comando.

3. Controlar o tempo de execução de cada processo com clock virtual.

Relógio Virtual:

1. Simular a passagem do tempo em ciclos (unidades de tempo virtual).

2. Controlar quando eventos devem ocorrer (fim de exec, E/S, etc.).

3. Servir de referência temporal para o Interpretador e Kernel.