Enunciado do projeto 2017 – PCS3446 – Sistemas Operacionais – Prof. João José Neto

Introdução.

São dadas algumas diretrizes para o projeto deste ano, cujo objetivo é o da elaboração de uma simulação da operação de um sistema operacional multiprogramado, a ser executado em uma versão modificada da MVN.

Para efeitos práticos, o que vem a seguir deve ser interpretado como uma proposta concreta de implementação, e não como uma especificação rígida a ser literalmente cumprida.

Isso significa que há uma grande liberdade para a introdução de alterações, desde que estas sejam relatadas em destaque, documentadas e devidamente justificadas tecnicamente.

Contudo, não se deve esquecer de que quaisquer alterações feitas na proposta, relativas a um ou outro dos quesitos do projeto, deverão preservar a consistência do projeto, uma vez que quase sempre implicam revisar todos os demais, pois são todos fortemente interdependentes.

Dessa forma, apresente no relatório os seus argumentos em favor da sua proposta, mostrando que, em seu conjunto, a integridade do projeto não se afeta pelas modificações introduzidas.

Proposta geral.

Deseja-se que o projeto atenda as seguintes especificações:

1. Deve ser utilizado um **motor de eventos** como base de implementação de **cada uma das simulações** guiadas por eventos, de que se constituirá o projeto.
2. Este projeto requer algumas **alterações na arquitetura da MVN,** incluindo:
   - Um **sistema de interrupções,** para viabilizar, entre outros, **entrada/saída básica** realística, **multiprogramação**, **virtualização de memória** e **chamadas de sistema**.
   - Criação do modo de **endereçamento indireto**, ausente na MVN original, para viabilizar a implementação do esquema de memória virtual aqui proposto.
   - Alteração do funcionamento de algumas instruções, eliminação de outras pouco usadas ou inúteis, e inclusão de instruções novas, adequadas a este projeto. Como não há códigos de operação vagos na MVN, **criar novas instruções requer desativar instruções existentes**, cedendo seu código para uma nova instrução. Naturalmente, convém desativar apenas instruções pouco utilizadas ou desnecessárias.
3. **Inclusão do conceito de ponteiros de 16 bits** para o cálculo do endereço efetivo de instruções de referência à memória em modo indireto de endereçamento:
   - Os dois bytes encontrados a partir do endereço apontado por uma instrução executada no modo indireto de endereçamento formam um **ponteiro de 16 bits**.
   - **Os 4 bits mais significativos** de um **ponteiro de 16 bits** selecionam **um dos até 16 processos** que podem compor o programa em execução.
   - **Os 12 bits menos significativos** desse **ponteiro** representam um **endereço** (interno ao particular processo selecionado pelos 4 bits iniciais do ponteiro). Este se interpreta como um endereço **relativo à posição física inicial de memória** a partir da qual o código deste processo foi carregado.
4. **Modificação do conjunto de instruções** para abrigar **endereçamento indireto**
   - Por ser facilmente substituível por qualquer instrução de desvio equivalente, **elimina-se a instrução RS,** para que seu código possa ser associado à **nova instrução IND**, que **transforma em indireto o modo de endereçamento de sua instrução seguinte** (de referência à memória). O conteúdo de memória apontado

pela instrução original é então interpretado não mais como operando, mas como um ponteiro de 16 bits para o operando.

- Com o auxílio do endereçamento indireto, torna-se possível criar programas com espaços de endereçamento grandes, em que os programas passam a ter acesso a até 16 bancos de memória, de 4K Bytes cada um, ampliando o alcance do endereçamento lógico dos programas para um máximo de 64K Bytes cada um.

5. Multiprogramação e extensão de memória física

- Embora nada impeça a implementação do recurso da multiprogramação, a disponibilidade de um espaço físico ampliado de memória favorece esta prática pela disponibilidade adicional de espaço físico na máquina.
- Isso possibilita compartilhar os bancos disponíveis de memória física entre os programas participantes da multiprogramação, que possam estar sendo executados simultaneamente.
- Neste caso, se o total de memória ocupada por todo o conjunto desses programas não superar o número de bancos físicos de memória disponíveis, é possível particioná-la, alocando-se a cada programa o número adequado de bancos de memória, e mapeando seus endereços lógicos em endereços físicos através de um mapeamento similar ao da alocação paginada de memória física.
- Para tal mapeamento, é necessário manter uma tabela de mapeamento de memória para cada um dos programas, associando a cada um dos seus bancos lógicos de memória o banco físico correspondente. Essa tabela deve ser consultada dinamicamente pelo hardware sempre que houver referência indireta a um banco de memória.

6. Memória virtual e suas interrupções

- É possível também, sem alterações adicionais substanciais à arquitetura da máquina, adicionar um esquema básico de virtualização de memória eliminando a restrição de que a área física total dos programas multiprogramados não exceda a memória física.
- Para tanto, os até 16 bancos físicos de memória disponíveis podem ser utilizados pelos programas envolvidos na multiprogramação via alocação sob demanda, e dessa forma, cada programa poderia endereçar até 16 bancos de memória lógica, e compartilhar os bancos de memória física com outros programas que estejam operando simultaneamente.
- Nesta proposta, a memória virtual opera sobre áreas de 4K Bytes cada um, a serem utilizados, a critério do programador, pelas instruções que formam o programa que está sendo executado.
- Para tanto, é necessário que esteja disponível no hardware do processador uma **interrupção especial**, cuja ocorrência deve se manifestar sempre que uma referência for feita a qualquer posição lógica de memória que não esteja no momento alocado em nenhum dos bancos físicos em uso na ocasião.
- Referências a posições lógicas devidamente alocadas na memória física devem ser mapeadas de acordo com o conteúdo da tabela de mapeamento do programa.
- Por outro lado, referências a posições lógicas que no momento não estejam alocadas na memória física, e que portanto não podem ser mapeados, exigem dos procedimentos de tratamento das interrupções por elas suscitadas.

- Uma das missões dessas rotinas de tratamento de interrupção é fazer com que o banco lógico referenciado seja devidamente alocado em um banco físico, e seu código, nele copiado.
- Naturalmente, copiar tal código em um banco físico de memória exige que se disponha de um banco físico livre. Caso isto não aconteça, é necessário que antes seja liberado um banco físico ocupado.
- Em qualquer dos dois casos, é necessário que tanto as tabelas de mapeamento de cada um dos programas envolvidos no processo como a tabela de mapeamento geral da memória física devem ser devidamente atualizadas a cada um desses tratamentos de interrupção.

7. Sistema de interrupção

O sistema de interrupções a ser incorporado à arquitetura do processador deve implementar um mínimo de funcionalidades dos sistemas de interrupção geralmente encontrados nos processadores usuais:

- Permite/Inibe interrupção - Um bit de estado indica se pedidos de interrupção podem ser atendidas ou não pelo processador. É necessário que haja uma instrução para ligar e desligar este bit. Ao iniciar o processamento, este bit deve estar desligado (interrupção inibida).
- Supervisor/Usuário - o modo "supervisor" é o modo de operação em que o sistema operacional opera: ao início do processamento e durante o atendimento de uma interrupção. Passa-se do modo supervisor para o modo usuário executando-se uma instrução de retorno de interrupção. Passa-se do modo usuário para supervisor: (a) quando se executa uma instrução de chamada de sistema; (b) quando a interrupção estiver permitida, algum pedido interrupção estiver ligado e o processador estiverem modo usuário;
- Um bit de estado do processador indica se está em curso ou não um tratamento de algum pedido de interrupção. Importante: interrupções são sempre tratadas em modo supervisor. Nesta proposta, só uma interrupção pode ser tratada de cada vez, ficando eventuais outros pedidos de interrupção aguardando até que o processador volte ao modo usuário antes que mais um deles possa ser atendido.
- O tratamento de interrupções é feito inicialmente passando o peocessador para modo supervisor, salvando o endereço de retorno na posição 0 do bloco 0 de memória, e desviando para o programa de atendimento da interrupção, que reside a partir da posição 2 do bloco 0 de memória. Este é também o endereço de partida do sistema, quando acionado pela primeira vez.

8. Entrada e saída e suas interrupções

- Dois bits de estado (em uso ou não; terminou ou não) são usados para cada dispositivo de entrada/saída: um para indicar se o dispositivo está sendo utilizado (ou seja, que uma operação está em curso no dispositivo) ou não, e outro, para indicar que o dispositivo já terminou a última operação iniciada, mas que o processador ainda não fez o respectivo tratamento.
- Pedidos de interrupção da parte dos dispositivos ocorrem quando os bits de estado do dispositivo estiverem ambos ligados ao mesmo tempo (ou seja, estando o dispositivo em uso, terminou a operação nele iniciada anteriormente).
- A rotina de tratamento de interrupção de entrada/saída deve desligar os dois bits de estado após tratar a interrupção, deixando o dispositivo livre e pronto para a sua próxima tarefa.

- As duas operações mais usuais correspondem às tarefas de leitura e de escrita. Dispositivos convencionais efetuam a entrada ou a saída de um byte por vez. O dispositivo deve ter à sua disposição um registrador de oito bits onde o dado que se está transferindo deve ser depositado.
- Se for dispositivo de entrada, o dado lido é depositado nesse registrador pelo dispositivo assim que estiver pronto, e deve ser retirado pela rotina de tratamento de interrupção para ser transferido para a memória do processador.
- Se for dispositivo de saída, esse dado deve ser colocado no registrador do dispositivo antes de o dispositivo ser acionado para efetuar a operação de escrita no meio externo.
- Em ambos os casos, a rotina de atendimento de interrupção deve deixar o dispositivo preparado para a próxima tarefa.

9. Chamada de supervisor e a interrupção correspondente

- Deve ser implementada a instrução de chamada de supervisor e pelo menos alguns de seus serviços essenciais.
- A execução de uma instrução de chamada de supervisor se resume essencialmente a provocar um pedido de interrupção, cujo atendimento deve tirar o processador do modo usuário, mudando seu estado para modo supervisor, salvando na posição 0 do bloco 0 de memória o endereço de retorno, e forçando a execução da rotina de tratamento geral de interrupção do sistema, presumidamente presente a partir da posição 2 do bloco 0.
- Implemente as chamadas de sistema essenciais, coletando as necessidades apresentadas neste texto. Um conjunto mínimo é o seguinte:
  - leitura e escrita em dispositivos de entrada/saída,
  - solicitação de final de processamento,
  - operações de create/delete, open/close, read/write aplicados a arquivos de texto.

10. Recomendações

- Para facilitar o trabalho, use o texto deste enunciado como uma lista de itens a incluir no relatório do trabalho realizado. Vá marcando os itens já incluídos até que todos tenham sido marcados.
- Não se esqueça de registrar e documentar todos os itens que forem modificados, justificando tecnicamente as alterações introduzidas.
- Não deixe de relatar detalhadamente os testes realizados e de interpretar os resultados obtidos.
- Entregue em um site google o relatório pdf (explícitos, sem compactação) e os arquivos do projeto completo e funcionando até a semana da segunda prova. Não compacte os arquivos.