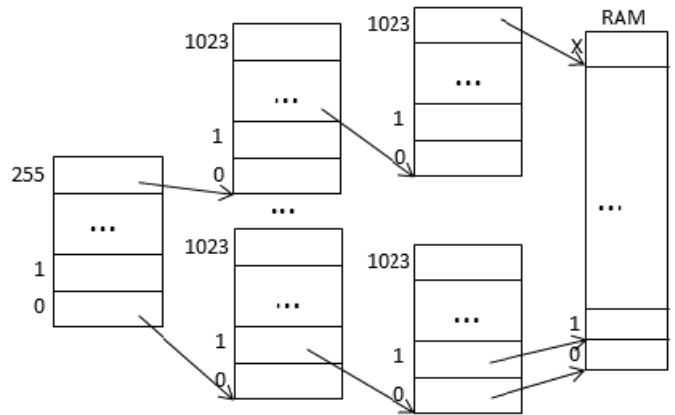


- 1) [3] Considere a organização da MMU apresentada à frente. Sabendo que cada PTE ocupa 4 bytes e que cada tabela de 2º nível ocupa exatamente uma página, indique, apresentando cálculos justificativos:

- a) [1] Qual a dimensão de cada página, e qual o número de *bits* de um endereço virtual?
- b) [1] Sabendo que o espaço de endereçamento físico é quatro vezes inferior ao espaço de endereçamento virtual, qual é o número total de *bits* do espaço de endereçamento físico, e quantos *bits* para controlo sobram na PTE?
- c) [1] Até quanto pode crescer o espaço de endereçamento virtual, mantendo o número de níveis apresentado para a MMU?



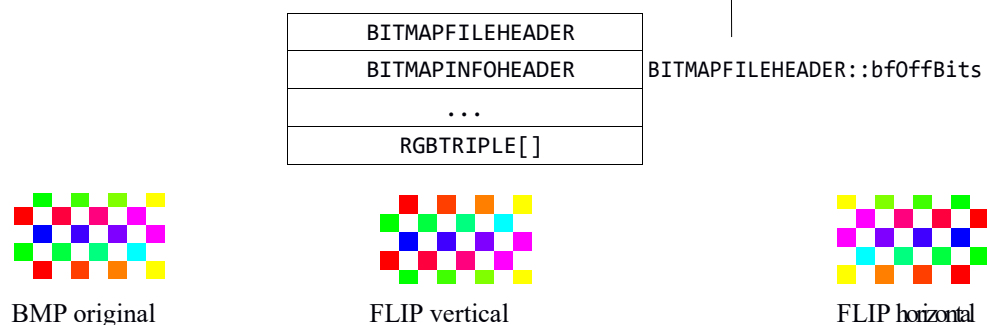
- d) Apresente um diagrama com as tabelas e as respectivas PTE envolvidas na tradução do endereço virtual 0x701053CC na página física que começa no endereço 0x400000. Apresente no seu diagrama os endereços físicos que considerou para as tabelas envolvidas na tradução.

- 2) Relativamente ao documento em anexo, que contém excertos da documentação sobre a arquitetura do processador Alpha, muito usado nos anos 90 em estações de trabalho avançadas, nomeadamente da HP, responda às seguintes questões:

- a) Embora um endereço virtual seja emitido a 64 bits, nem todos os bits são usados no processo de tradução. Qual o número mínimo de bits usado na tradução de endereços, nas possíveis implementações da arquitetura?
- b) A arquitetura suporta implementações com diferentes dimensões de página, embora mantendo o número de níveis de tradução de endereços e a dimensão de cada entrada (PTE) das tabelas de páginas. Quantos níveis de tradução presentes e qual a dimensão em bits da PTE, qualquer que seja a dimensão de página? Em relação à PTE, qual o número de bits usados para controlo e para especificar a *page frame*?
- c) A escolha da dimensão da página, para uma dada implementação, tem implicações diretas no espaço de endereçamento virtual. Para cada dimensão da página (e tabelas de páginas), quais os intervalos possíveis para o número de bits do espaço de endereçamento virtual? Indique também, para cada caso, o número de entradas ocupadas na tabela de 1º nível.
- d) Um aspeto curioso da arquitetura é a organização em regiões (denominadas segmentos) do espaço de endereçamento virtual. Indique quais as regiões existentes, o nível de privilégio de execução necessário para o acesso a cada região, e o intervalo de endereços válidos para cada região. Para este último valor, considere um espaço de endereçamento de 43 bits.
- e) Numa das regiões indicadas na alínea anterior, os bits efetivos do endereço virtual (retirando os dois bits mais significativos) que identificam a região, correspondem diretamente ao endereço físico. Sugira possíveis utilizações para esta região. Associe a existência desta região ao facto do máximo espaço de endereçamento físico ser sempre inferior à dimensão do espaço de endereçamento virtual, para uma dada implementação. Que outro fator concorre para a limitação do espaço de endereçamento físico?

- 3) Faça um diagrama de estados que apresente os estados por que passa uma *page frame* no Windows. Considere como estados iniciais e finais do ciclo de vida aqueles em que a *page frame* se encontra completamente dissociada de qualquer espaço de endereçamento (*free* ou *zeroed*). Dê nomes às transições entre estados, descrevendo pelo menos um cenário ilustrativo de cada transição.

- 4) Utilizando a função de sistema `QueryWorkingSet`, presente na biblioteca `psapi`, realize uma aplicação que, recebendo o *id* de um processo arbitrário, determine se o processo aparenta ter *leaks* de memória. Para isso a aplicação determina periodicamente (ex: de 5 em 5 seg.) o total de espaço físico (*working set*) de regiões de dados privados do processo a analisar (para a temporização utilize a função `Sleep` da API do Windows). Se o total ultrapassar determinado *threshold* pré-definido, o programa emite mensagem para o *standard output* informando do facto. Experimente com programas de teste.
- 5) Escreva um programa que realiza uma operação de rotação de 180° (*flip*), sobre um ficheiro BMP segundo os eixos, vertical ou horizontal. O programa recebe por parâmetro dois nomes de ficheiros e um eixo de rotação: o primeiro nome indica o ficheiro BMP original, o segundo nome identifica o ficheiro a criar com o resultado da operação e o eixo de rotação identifica o eixo (vertical ou horizontal) sobre o qual a rotação será aplicada. O acesso ao ficheiro BMP deve ser realizado exclusivamente por mapeamento em memória. As figuras seguintes apresentam um resumo da estrutura de um ficheiro BMP com 24 bits por pixel, uma imagem num ficheiro BMP e as imagens com os respectivos resultados das duas operações de rotação.



Notas:

- O programa processa apenas ficheiros *bmp* com 24 bits por pixel (`BITMAPINFOHEADER::biBitCount == 24`) não comprimidos e que possam ser totalmente mapeados no espaço de endereçamento disponível do processo.
- Tome em atenção que os pixels estão organizados por linhas que, independentemente da dimensão em colunas, se encontram alinhadas em endereços múltiplos de 4. Poderá encontrar mais informações sobre o formato *bmp* em <http://www.tinaja.com/glib/expbmp.pdf>.
- Com a ferramenta *Paint* pode abrir ficheiros *JPEG* e salvá-los como *BMP* de 24 bits.

Use o programa da questão 4 para analisar a memória utilizada por este programa durante o processamento de um ficheiro *BMP*. Analise e comente os resultados obtidos nos seguintes momentos: antes da criação do *Section Object*, antes do mapeamento dos ficheiros no espaço de endereçamento do processo, antes de qualquer acesso aos pixels dos ficheiros e depois da conclusão da operação.

- 6) Implemente uma *DLL* que exporta a seguinte função:

```
VOID BMP_Flip(TCHAR filenameIn, TCHAR filenameOut, FLIP_enum_t type);
```

A função é uma adaptação do programa da alínea anterior. A sua implementação deve suportar tanto clientes que usam caracteres codificados no código *ASCII* como clientes que usam caracteres codificados no código *Unicode*.

Data limite de entrega: 9 de Abril de 2018

João Patriarca, João Trindade e Jorge Martins
ISEL, 7 de Março de 2018