



Exercício 1

a)

O espaço de endereçamento está dividido em três níveis de tabelas de páginas, como este espaço é de 256GiBytes sabemos que vamos distribuir 38 bits (2^{38} bytes). Queremos idealmente que o *offset* tenha o menor tamanho possível em adjacência com ter um número igual de bits para cada tabela no espaço de endereçamento virtual.

Se escolhermos um *offset* de 10 bits, então o tamanho das tabelas vai ser 2^{10} bytes (1k), e ficávamos com 28 bits restantes que não são divisíveis por 3, ou seja iam sobrar bits no espaço de endereçamento se quisermos atribuir um número igual a cada tabela. Se usarmos 11 bits de *offset*, então já conseguimos distribuir os restantes 27 bits pelas três tabelas de igual forma:

9 bits

9 bits

9 bits

11 bits = 38 bits

b)

Como são utilizados 6 bits para as *flags* e a PTE tem entradas de 4 bytes (32bits), ficamos com 26 bits para o endereçamento, juntando com os 11 bits de *offset*, ficamos com os 37 bits indexáveis.

Ao adicionar 1 bit de controlo, são utilizados 7 bits para *flags* o que reduz o espaço de endereçamento.

c)

A afirmação é verdadeira só quando se acede à memória através de um endereço que nunca foi acedido anteriormente (ou seja, não está armazenado na cache). Ao existir uma cache associada à arquitetura, podemos evitar acessos físicos, caso esta não exista é inevitável.

Exercício 2

a)

As listas *standby* e *modified* servem para quando uma página é retirada de um processo mas é guardada na memória principal (*page buffering*), caso o processo necessite da mesma página, esta é devolvida imediatamente ao processo.

No caso da lista *standby*, a página não teve nenhum acesso de escrita (bit *dirty* a 0), porém a lista *modified* contem as paginas marcadas com o bit *dirty* a 1, ou seja, tiveram acessos de escrita.

b)

Quando o Windows inicia, todas as páginas estão na lista *free*, e à medida que os processos utilizam memória do sistema, as páginas das listas *free* e *zero* vão sendo consumidas, mas quando um processo termina ou liberta páginas, estas vão para as listas *standby* e *modified* em vez de voltar às *free* e *zero*. Ou seja, as listas *free* e *zero* não aumentam à mesma velocidade que diminuem.

c)

A função VirtualFree liberta um conjunto de páginas dentro do endereçamento virtual do processo. Se a função for chamada com o parâmetro *dwFreeType* igual a `MEM_RELEASE`, então as páginas libertadas vão diretamente para a lista *free*.

Exercício 3

(código fornecido em anexo)

Exercício 4

a)

A vantagem é impedir que as DLL sejam relocizadas.

b)

Quando ocorrem interseções entre duas DLL, o loader reloca uma delas ao utilizar a secção `.reloc`, que possui todas as entradas de DLL relocizadas.

Exercício 5

(código fornecido em anexo)