|  |
| --- |
|  |
| Sistemas Operativos |
|  |
| LI41d |

André Martins

André Gaudêncio

Leonardo Freire

Sistemas Operativos

LI41d

## Exercício 1

1. Tendo em conta que uma página possui no máximo uma capacidade de 212 e que cada página possui entradas de 8 bytes

Assumindo que os níveis possuem todos o mesmo tamanho

212=23x?

?=212/23=29

Tendo em conta que existem 3 niveis necessitamos de 27 bits para descobrir os níveis. Assim o endereço virtual possui no máximo 64-27-12=25 bits livres.

1. Se cada entrada das tabelas de páginas armazenar 38 bits do endereço físico, e admitindo que esses 32 bits da PTE são para a base do endereço, então existem 250 endereços. Logo a memoria tem dimensão máxima 1 petabyte.
2. A organização dos endereços virtuais:

|  |  |  |  |
| --- | --- | --- | --- |
| 1ª Tabela | 2ª Tabela | 3ª Tabela | Offset |
| 6 bits | 9 bits | 9 bits | 12 bits |

12+9=21

21+9=30 +6=36 bits

1. Considerando os valores das tableas
   * 1. Sasa

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Posiçoes de memoria | offset | 1º Nível | 2º Nível | 3º Nível |
| 0x430470 | 0100 1111 0000 | 000000 | 000000010 | 000110000 |
| 0x4325E8 | 0101 1010 1000 | 000000 | 000000010 | 000110010 |
| 0x22100 | 0001 1100 0000 | 000000 | 000000000 | 000100010 |

* + 1. Ao ser retirada a permissão de escrita

## Exercício 2

1. 1.A pagina referida não está submetida, ou seja, o programa tentou aceder a um endereço que não tem mapeamento valido (por exemplo, se o endereço já foi libertado). Geralmente o handler termina o processo.

2.Acesso a uma pagina sem as devidas permissões, ou seja, um processo tenta realizar uma operação invalida, como o exemplo de escrever numa pagina que apenas permite leitura. Novamente o programa é terminado.

3.O problema é parecido com o 2., ou seja, o processo tenta realizar uma operação invalida sobre uma pagina, mas desta vez o handler resolve o problema de forma diferente, pois agora a pagina estava marcada como copy-on-read, e assim o memory manager não assume que houve uma violação de acesso, e faz ao invés disso uma copia privada da pagina ao processo atual. Quando o processo volta ao controlo da pagina e tenta fazer novamente a operação, esta já executada de forma normal.

4.Quando uma thread insere um valor no seu stack e cruza com uma pagina que ainda não foi inicializada ou alocada, porem o memory manager esta preparado para detetar esta situação e reconhece como um caso especial, e o que faz é disponibilizar uma nova pagina, colocar todos os seus valores a 0, e mapeá-la ao processo, isto desde que ainda exista espaço nas paginas virtuais para o stack.

5.Falha normal na pagina. Existem vários tipos de falhas normais nas paginas e diferentes resoluções consoante o tipo de erro.

1. A alternativa da Microsoft é a seguinte
2. Uma soft fault é quando o memory manager consegue resolver o problema acedendo à página em memória, sem ser necessário voltar a ler a página a partir do disco, no caso de ser necessário aceder ao disco para recuperar a página estamos perante uma hard fault.

As soft fault segundo o autor podem ocorrer por causa de uma página partilhada já estar mapeada a um processo, ou quando é necessária uma página a zero. Também pode ocorrer quando uma página é comprimida para aumentar o tamanho da memória física.

1. A introdução do swap file nos sistemas operativos modernos foi introduzido para que as aplicações do WUP (Windows Universal Program) quando não estiverem a ser utilizadas em foreground são suspensas, e não executam até o utilizador voltar a selecionar essa aplicação. Quando o sistema necessita de memória, é selecionada uma aplicação que não esteja a ser usada há algum tempo e é feito o swap de todas as páginas em várias operações IO ao invés de ser feito página a página. Depois quando a aplicação voltar a ser executada é feito uma leitura sequencial o que é mais eficiente que random reads.

A motivação para o aparecimento do swap file foi o facto de otimizar o uso de recursos do sistema, aumentando a duração da bateria, visto que as aplicações cada vez usam mais recursos e mais aplicações são usadas em simultâneo.

1. A partir do Windows 8 é utilizada uma funcionalidade chamada Physical Address Extension(PAE) que define uma hierarquia de tabelas de pagina de 3 níveis, com entradas de tabela de 64 bits em vez dos tradicionais 32 bits, permitindo ao processador aceder a endereços físicos num espaço físico superior aos 4 GB. Assim o autor do texto não está a cometer um erro ao dizer que as PTE são de 8 bytes.

### Exercício 4

1. A vantagem de as bibliotecas do sistema ocuparem os mesmos endereços independentemente dos processos em observação é a de permitir uma maior eficiência no tempo de carregamento e na gestão da memória.
2. O loader resolve esta questão procedendo a realocação em tempo de carregamento das dlls para outros endereços de memória. Para evitar esta situação de realocar a memória no tempo de carregamento é possível utilizar uma ferramenta *rebase* para mudar os endereços antecipadamente e evitar as interceções.