

**Relatório 1ª Serie de SO**

Docente: João Pedro Patriarca

Semestre Verão – 16/17

Turma LI42D

**Alunos:**

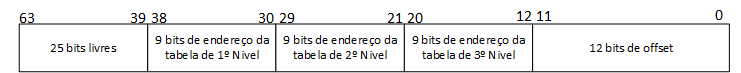
Rui Franco - 36210

Ana Baptista - 41487

Miguel Arroja - 42161

**Exercício 1:**

a)



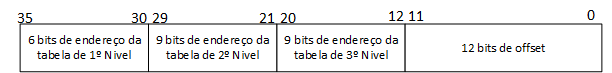
-4 KiB -> ->12 bits de offset

-Endereço da tabela : = = -> 9 bits de endereço

b) 38 bits do endereço físico presente na tabela+ 12 de offset = 50 bits de endereço físico

-> 1 PetaByte de memoria Física

c)



**Exercício 2:**

a)

1)A trap to the kernel occurs.

2)The kernel builds a machine-independent descriptor telling what happened.

3)The kernel passes the descriptor to the memory-manager part of the executive. 4)The memory-manager checks the access for validity.

5)case the faulted page falls within a commited region, it looks for the address in the list of VADs and finds (or creates) the process page-table entry. case the faulted page be a shared page, the memory manager creates a copy of it, being now allowed to be edited.

b) Prepaging – As páginas referentes não entram diretamente na tabela de páginas do processo, mas sim para o standby list, em que poderão ser utilizadas ou não pelo processo. A Microsoft chama-lhe de *Super Fetch*. Em que este método tenta prepaginar muitas das páginas necessárias, ainda que não tenham sido necessárias. Reduz a latência no inicio de cada processo. É também mais fácil carregar estas páginas, visto que estão no disco e o tempo de leitura é reduzido.

c) Segundo o autor, o que leva a um soft fault é quando um processo tenta aceder a informação que anda está em memória principal, mais concretamente em na lista de stand by.

d) *Swap file* consiste em garantir mais memória a processos que tenham estão a ser utilizados como principais, ou seja, que não estão em segundo plano. O espaço de memória para estes últimos é limitado, em que poderão ser substituídos por espaço de memória dos processos principais. Esta técnica foi desenvolvida para realizar menos operações I/O ou seja, de acessos a memória secundária.

**Exercício 4:**

a) Processos escolhidos:

Notepad ++ : kernel32.dll -> 0x76D60000 ; user32.dll -> 0x76C00000

Adobe Reader : kernel32.dll -> 0x76D60000 ; user32.dll -> 0x76C00000

Através do VMap verificamos que ambas as dll’s partilham o mesmo endereço em ambos os processos. Desta forma podemos observar a memoria partilhada do Windows entre processos, sendo que as paginas destes processos são marcadas como Read-Only e Copy-on-Write.

b) O loader procede à realocação das mesmas para outros endereços (isto ocorre em tempo de carregamento). Para evitar esta realocação em tempo de carregamento é possível utilizar o utilitário rebase para mudar os endereços antecipadamente e assim evitar intersecções.