Instituto Superior de Engenharia de Lisboa

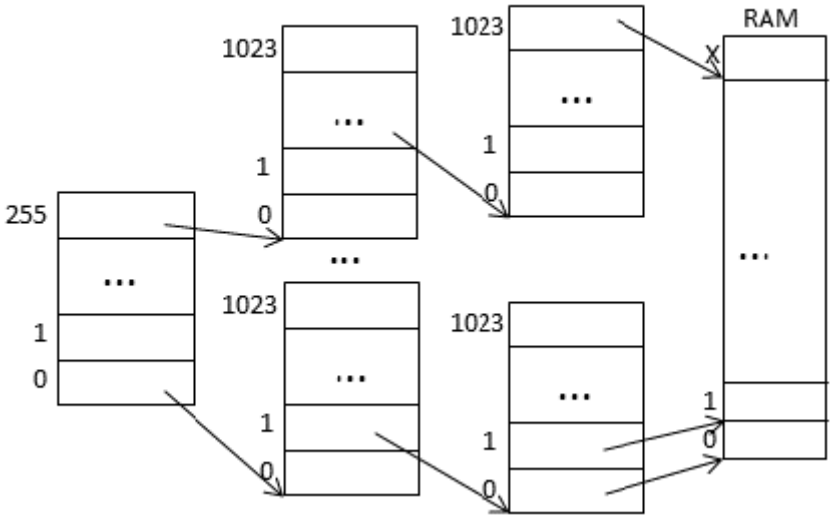
Licenciatura em Engenharia Informática e de Computadores

Sistemas Operativos, Verão de 2020/2021

Segunda série de exercícios

---

1. Considere a organização da MMU apresentada abaixo. Sabendo que cada PTE ocupa 4 *bytes* e que cada tabela de 2º ou 3º nível ocupa exatamente uma página, responda às questões seguintes, apresentando os cálculos justificativos.



   a. Qual a dimensão de cada página e qual o número de *bits* de um endereço virtual?

   b. Sabendo que <u>o espaço de endereçamento físico é quatro vezes inferior ao espaço de endereçamento virtual</u>, qual é o número total de *bits* do espaço de endereçamento físico, e quantos *bits* para controlo sobram na PTE?

   c. Até quanto pode crescer o espaço de endereçamento virtual, mantendo-se o número de níveis apresentado para a MMU?

   d. Considere a tradução do endereço virtual 0x701053CC para o respetivo endereço físico localizado na página física que começa no endereço 0x400000. Estabeleça um endereço físico arbitrário para a base da tabela de primeiro nível, ficando as restantes tabelas envolvidas no processo de tradução em endereços físicos contíguos crescentes. Apresente a sequência de passos necessários para realizar a tradução do endereço virtual, indicando em detalhe:
      - o endereço base de cada tabela de tradução utilizada
      - o endereço de cada PTE envolvida na tradução
      - os bits de endereço físico contidos em cada uma dessas PTE
      - o endereço físico final

   e. Comente a seguinte afirmação: "Nesta arquitetura, cada acesso lógico à memória, correspondente a uma operação de *fetch*, *load* ou *store* do CPU, implica sempre quatro acessos físicos à memória."

NOTA: Coloque as respostas ao exercício 1 no ficheiro `se2/ex1/ex1.md` no repositório de grupo.

2. Construa um programa que recebe como argumento o nome de um ficheiro e que escreve nesse ficheiro uma linha com o *process id* da execução do programa e uma linha por cada elemento do grupo, contendo os respetivos números e nomes. Por exemplo, invocando `prog se2-ids.txt` o ficheiro `se2-ids.txt` fica com:

```
3312
11111 - Afonso Henriques
12345 - Alexandre Herculano
13579 - Fernando Pessoa
```

A obtenção do *process id*, bem como a abertura, escrita e fecho do ficheiro devem ser feitas por rotinas em *assembly* que realizam diretamente as chamadas de sistema a *getpid*, *open*, *write* e *close*, recorrendo à instrução *assembly* `syscall`.

NOTA: Coloque a resolução do exercício 2 na diretoria `se2/ex2/src` do repositório de grupo, nos ficheiros `prog.c` e `syscalls.s`, com o respetivo `makefile` e um ficheiro *header*, se for conveniente.

Entrega

Coloque as respostas ao exercício 1 no ficheiro `se2/ex1/ex1.md`

Coloque a implementação do exercício 2 na diretoria `se2/ex2/src/` do repositório de grupo, com o ficheiro fonte `prog.c` contendo o programa em C e o ficheiro `syscalls.s` contendo as rotinas `xgetpid`, `xopen`, `xwrite` e `xclose` em *assembly* x86-64, acompanhados do respetivo `makefile`.

A entrega é finalizada usando a *tag* **SE2** no repositório GitHub.

ISEL, 17 de abril de 2021

Data limite de entrega: 26 de abril de 2021