Instituto Superior de Engenharia de Lisboa

Licenciatura em Engenharia Informática e de Computadores

Técnicas de Virtualização de Sistemas, Inverno de 2023/2024

Época Normal — Teste Parcial #1 — Teste Global, parte 1

ATENÇÃO: Responda às questões **1** e **2** num conjunto de folhas e às questões **3**, **4** e **5** noutro conjunto.

1. [6] Considere o código que será compilado para o executável **oddlines** e, à direita, o ficheiro **in.txt**:

```
int main(int argc, char * argv[]) {
  int nl = 1;
  char line[MAXLINE];
  while (fgets(line, MAXLINE, stdin)) != NULL) {
    if (nl % 2 != 0)) {
      fputs(line, stdout);
    }
    nl++;
  }
  return 0;
}
```

```
ISEL, Lisboa
IST, Lisboa
ISEP, Porto
ISCTE, Lisboa
```

   a. [1.5] Justifique o conteúdo do ficheiro out.txt com a execução da seguinte linha de comando:

      `oddlines < in.txt | grep Lisboa > out.txt` ← *mesmo comando para as alíneas **b** e **c***

   b. [1.5] Para a linha de comando acima, quantos processos são criados e, para cada um deles, o que referem as entradas 0 e 1 das respectivas tabelas de descritores de ficheiros?

   c. [3] Escreva um programa em C que execute especificamente a linha de comandos indicada na alínea ***a***, usando diretamente a API POSIX para criar/abrir ficheiro(s), lançar os processos necessários, ligá-los da mesma forma que o *shell* e aguardar pela sua conclusão.
   NOTA: *Usar, direta ou indiretamente, uma instância de shell terá uma classificação de zero.*

2. [4] Em 2017, a Intel propôs uma extensão à arquitetura *x86-64*, para incluir um quinto nível de tabelas no esquema de tradução de endereços virtuais. Com esta extensão, os endereços virtuais continuam a ter 64 bits mas a tradução de um endereço implica trabalho adicional, devido ao quinto nível de tabelas.

   a. [2.5] Explique o que mudou na estrutura dos endereços virtuais e qual a vantagem de adicionar o 5º nível de tradução, apesar do custo adicional que implica.

   b. [1.5] Que elemento da arquitetura da unidade de gestão de memória (MMU) do CPU é essencial para suavizar o custo da tradução de endereços (seja a 4 ou 5 níveis) e porque é, em geral, eficaz?

3. [3.5] Considere o seguinte código fonte de um programa em C:

```
1  const int year = 2024;
2  char data[16384] = { 0, 1, 2, 3, 4, 5, 6, 7 };   // remaining 16376 will be 0
3  void main(int argc, char * argv[]) {
4    char * ptr = malloc(4 * 1024 * 1024);        // allocate 4MB
5    memcpy(ptr, data, 16384);                   // copy 16KB from data into ptr
6    void * lib = dlopen("./libtp2.so", RTLD_NOW);
7    int * p = (int*)&year;   *p = 2025;       // advance to next year...
8  }
```

Para cada linha de código da função `main` (4 a 7), justifique que alterações prevê que acontecerão à organização do espaço de endereçamento do processo, com foco **apenas** na adição, remoção ou mudança de tamanho das regiões de endereços virtuais válidos, visíveis em `/proc/<pid>/maps`.

4. [2.5] Considere um sistema operativo Linux em execução numa máquina com processador da arquitetura *x86-64*, bem como outros tipos de sistemas semelhantes. Indique, em detalhe, três situações **concretas e distintas** em que ocorre transição da execução de modo utilizador para modo *kernel*.
   DICA: a iniciativa para desencadear a transição pode ser do *software* ou não e pode resultar de erro ou não.

5. [4] O seguinte programa em C foi compilado e executado num sistema Linux com processador *x86-64* :

```
1  char u[5000];
2  char x = 'a';
3  char y;
4  int main(int argc, char * argv[]) {
5     y = x;
6     getchar();
7     return 0;
8  }
```

   a. [2] Na execução da linha 5, correspondente a uma instrução *assembly* `movb`, que região ou regiões de memória estão envolvidas e que mudança(s) se observa(m) no seu *resident set size* (`Rss`)?

   b. [2] Tendo a execução ficado parada na linha 6, executou-se outra instância do mesmo programa, noutro terminal, que também pára na linha 6. Em ambos os processos (do mesmo programa), o *resident set size* (`Rss`) da região de memória correspondente à secção `.text` vale 4KiB e o da região correspondente a `.data` é de 8KiB.

      i. [1] Quanto vale o *proportional set size* (`Pss`) nessas regiões `.text`? Justifique.

      ii. [1] Sabendo que o `Pss` das regiões `.data` era de 6KiB quando teve início a execução da função `main` no segundo processo, justifique quantas páginas de memória física estão partilhadas por estas regiões, entre os dois processos, quando ambos param na linha 6.

Duração: 1 hora e 15 minutos

ISEL, 17 de janeiro de 2024