Instituto Superior de Engenharia de Lisboa

Licenciatura em Engenharia Informática e de Computadores

Técnicas de Virtualização de Sistemas, Inverno de 2023/2024

Época de Recurso — 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 **tag2err** e, à direita, o ficheiro **in.txt**:

```
int main(int argc, char * argv[]) {
  int tlen = ((argc >= 2) ? strlen(argv[1]) : 0);
  char line[MAXLINE];
  while (fgets(line, MAXLINE, stdin)) != NULL) {
    if (strncmp(line, argv[1], tlen) == 0) {
      fputs(line, stderr);   // tag matched
    } else {
      fputs(line, stdout);   // tag not matched
    }
  }
  return 0;
}
```

```
INFO: operation started
INFO: id is valid
WARN: amount >= 10000
Server execution failed
Server detected an error
ERR: operation failed
INFO: retrying operation
Server rejected retry
ERR: retry failed
```

   a. [1.5] Justifique o que se observa no terminal e qual conteúdo do ficheiro `out.txt` com a execução da seguinte linha de comando:

      `grep operation < in.txt | tag2err ERR  > out.txt` ← *também para as alíneas **b** e **c***

   b. [1.5] Para a linha de comando acima, quantos processos são criados pelo *shell* 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 **mesmos** processos que o *shell* lançaria, 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] Num sistema Linux com processador *x86-64*, um processo tem o espaço de endereçamento configurado para que os seguintes endereços virtuais sejam acessíveis em *user mode* com as permissões indicadas ao lado:
   - 0x0003579BDF1569BD r__ *(Read OK , Write NOK , Execute NOK)*
   - 0x0003579BDF356543 rw_ *(Read OK , Write OK , Execute NOK)*
   - 0x0003579BDF357531 r_x *(Read OK , Write NOK , Execute OK)*

   a. [2] Indique o índice de cada *page table entry* acedida em cada um dos níveis da tradução, para o **primeiro** endereço da lista acima. Apresente os cálculos.

   b. [2] Quais são as permissões disponíveis em *user mode* para os seguintes endereços? Justifique.

      0x0003579BDF3569BD 0x0003579BDF357543 0x0313579BDF357531

3. [2.5] Os sistemas operativos da família Linux, bem como outros semelhantes, precisam que exista a distinção entre dois modos de execução: o modo *kernel* e o modo utilizador.

   a. [1.5] Em que se distinguem estes dois modos e que relevância têm para o sistema operativo.

   b. [1] Escolha **um dos dois** processadores utilizados como referência (*x86-64* ou *ARM64*) e indique que estado do processador escolhido corresponde a cada um dos modos.

4. [3.5] Considere o seguinte código fonte de um programa em C:

```
 1  char * mem1;  char * mem2;
 2  int main() {
 3    mem1 = mmap(NULL,16,PROT_READ|PROT_WRITE,MAP_PRIVATE|MAP_ANONYMOUS,-1,0);
 4    mem2 = mmap(NULL,256,PROT_READ|PROT_WRITE,MAP_SHARED|MAP_ANONYMOUS,-1,0);
 5    strcpy(mem1, "ISEL"); strcpy(mem2, "LEIC");
 6    if (fork() == 0) {
 7      strcpy(mem1, "DEETC"); strcpy(mem2, "TVS"); munmap(mem2, 256);
 8      mem1 = mmap(NULL,8,PROT_READ|PROT_WRITE,MAP_SHARED|MAP_ANONYMOUS,-1,0);
 9      strcpy(mem1, "Lisboa");
10    } else {
11      wait(NULL); // wait for the only child
12      puts(mem1); puts(mem2);
13      munmap(mem1, 32);
14    }
15    return 0;
16  }
```

   a. [1.5] Indique as linhas onde ocorrem mudanças na organização do espaço de endereçamento dos processos envolvidos. Em cada caso, indique o processo, a instrução que o afeta e qual a alteração. Considere apenas as mudanças visíveis na lista de regiões em `/proc/<pid>/maps` e não o RSS.

   b. [1] Aponte duas instruções onde ocorre *copy-on-write* e quais são as respetivas regiões. Justifique.

   c. [1] Indique, justificando, o que é escrito pelas instruções da linha 12.

5. [4] Em cada um dos casos seguintes, indique e justifique brevemente qual é o *backing storage* para as páginas físicas de memória referidas.

   a. [1] Página pertencente ao mapeamento da secção de constantes (`.rodata`) de um executável.

   b. [1] Página modificada, no mapeamento da secção de dados não-inicializados (`.bss`) de um ficheiro executável.

   c. [1] Página não-modificada, pertencente ao mapeamento da secção de dados inicializados (`.data`) de uma biblioteca (ficheiro com extensão `.so` ou *shared object*).

   d. [1] Página modificada, pertencente ao mapeamento SHARED de um ficheiro comum.

Duração: 1 hora e 15 minutos

ISEL, 29 de janeiro de 2024