UNIVERSIDADE DE AVEIRO

DEPARTAMENTO DE ELECTRONICA, TELECOMUNICAÇÕES E INFORMÁTICA Teste de Sistemas Operativos 16/janeiro/2023

NOTE BEM: Justifique todos os passos das suas respostas. Respostas não justificadas não serão consideradas.

Considere o seguinte código

```
1. int main(int argo, char *argv[])
        printf("1\n"); =01
  3.
        fork () :
       if(fork()!=0)
         printf("2\n");
          execlp("echo", "echo", "3", NULL);
         printf("4\n") ; > Nuce i emiliate
8.
0
      printf ("5\n");
10.
      return 0;
11.
12. 1
```

- a) Quantos processos, incluindo o inicial, são criados pela execução do programa anterior, (supondo que forks e execs não falham)? Desenhe um diagrama em árvore que descreva a relação entre os vários processos que são executados e as operações que estes executam. Atribua um número PID possível a cada ramo da árvore (que representa parte do fluxo de execução de um processo).
- b) Indique quais as linhas do código anterior que são executadas pelo processo principal e pelo(s) processo(s) que é(são) criado(s) pelo processo principal. Pode usar os PIDs do diagrama anterior
- c) Supondo que todos os forks e execs têm sucesso, apresente duas saldas possíveis (com conteúdo distinto) que podem resultar do código anterior.
- d) Desenhe o diagrama de estados de um processo e descreva sucintamente cada estado.
- Considere que o código apresentado à esquerda é executado por todas as threads que 11. pretendem executar READ access () e o código à direita por todas as que pretendem executar WRITE access (). Estes códigos usam uma variável partilhada (readers) e 2 semáforos partilhados por todas as threads: mutex e nobody (ambos inicializados a 1).

1. //readStart 2. mutex.down() 3. if(readers==0) 4. nobody . down () 5. readers++ 6. mutex.up() 7. READ access () 8. //readEnd 9. mutex.down() 10 readers --11. if (readers==0) 12. nobody up () 13. mutex , up ()

1. //writeStart 2. nobody.down() 3. WRITE access() 4. //writeEnd 5. nobody.up()

a) Supondo que existe uma Ihread que está a executar write_access(), explique o que acontece a todas as threads que tentarem executar o código da esquerda. Indique claramente a posição de todas as situações de bioquelo/desbloqueio.

- b) Explique porque pode o código anterior implicar o adiamento indefinido das threads escritoras. Altere o código anterior, de modo a evitar a o adiamento indefinido das threads escritoras. Dica: use mais um semáforo (ex: turnstile)
- c) Indique quais as 4 condições necessárias à ocorrência de deadlock. Usando semáforos crie o código de 2 processos em que o deadlock possa ocorrer
- III. Considere um sistema de ficheiros FAT32 em que os clusters 0 e 1 estão reservados e que as primeiras 16 entradas têm os valores apresentados na figura (0000 0000 livre, FFFFFFF final)

۰	11).				PECC CEEE			
					0000 0008	FFFF FFFF	0000 0000	0000 0000
	VVVV VVVV	YYYY XXXX	0000 0005	0000 0004	0000 0000		0000 0000	0000 0000
d	YYYY YYYY	AAAA AAAA		CCCC FFFF	0000 0000	FFFF FFFF	0000 0000	
1	0000 0007	9000 000B	0000 0000	FEFF				

- a) Sabendo que a diretoria raiz está colocada a partir do cluster 2, e que existem 3 ficheiros que têm como primeiro cluster os clusters 3, 9 e 13, respetivamente. Indique quais os clusters que constituem a diretoria raiz e cada um dos clusters
- b) Sabendo que cada cluster tem 1024 bytes, indique qual a posição no disco (cluster e offset), do byte 1200 (1° byte do ficheiro é o byte 0) do ficheiro que se inicia no cluster 3
- Considere que os processos P1, P2, P3 e P4 têm um tempo total de execução de 100ms, IV. 50ms, 10ms e 45ms, respetivamente.
- a) Apresente o diagrama temporal do escalonamento destes processos usando o algoritmo Round Robin com um quantum time de 20ms.
- b) Qual o tempo médio de espera dos processos para o escalonamento da alínea anterior?
- c) Apresente as vantagens e desvantagens do escalonador Round Robin relativamente ao escalonador FCFS. Indique também, justificando, qual o mais indicado para escalonar aplicações interativas e qual o mais indicado para escalonar aplicações de cálculo intensivo.
- O conteúdo da memória e a tabela de página de cada processo ativo de um sistema de V. memória virtual paginada são apresentados a seguir (de forma simplificada, considerando que cada página tem o tamanho de 1 byte e que apenas existem 2 processos):

P1	Memória	Tabela de página	P2 ,	Tabela de página	
0:		2	0:	E	9
1:	В	7	1:	F	0
2:	С	1	2:	G	12
3:	D	3	3:	Н	15

- a) Apresente o conteúdo da memória física entre os endereços 0 e 15
- b) Explique o conceito de page fault. Refira também como é detetado um page fault e quais os passos envolvidos na sua resolução.
- c) Quais os principais objectivos da Memória Virtual?