

Exame de Sistemas Operativos – Perguntas

Verdadeiro/Falso

1. A variável \$PATH define onde o sistema procura executáveis. (V/F)
2. O comando `test` e `[]` têm o mesmo comportamento no Bash. (V/F)
3. A chamada fork() cria um novo thread dentro do mesmo processo. (V/F)
4. Um ficheiro binário aberto com fopen(...,"rb") não permite escrita. (V/F)
5. Um semaphore POSIX inicia sempre com valor 0. (V/F)
6. A função exec substitui o processo atual por um novo programa. (V/F)
7. pthread_join() é usado para criar threads. (V/F)
8. A memória partilhada desaparece automaticamente quando o último processo a fecha. (V/F)
9. O comando `ls -l` mostra permissões, owner e grupo. (V/F)
10. A função wait() permite recolher o exit code de um processo filho. (V/F)

Escolha Múltipla

11. Qual comando imprime as variáveis de ambiente?
A) printenv B) echoenv C) envls D) whoami
12. Qual opção abre um ficheiro para escrita sem truncar?
A) "w" B) "r+" C) "a" D) "rw"
13. Depois de fork(), qual variável é diferente no filho?
A) PID B) Stack C) Código D) Dados
14. Qual chamada cria um thread?
A) fork() B) pthread_create() C) exec() D) newthread()
15. Semáforos são usados para:
A) memória virtual B) sincronização C) compressão D) pipes

Correspondência

16. Associe:

- (1) fork()
 - a) Sincronização
 - b) Substitui o processo
 - c) Espera por filho
 - d) Duplica processo
- (2) exec()
 - a) Sincronização
 - b) Substitui o processo
 - c) Espera por filho
 - d) Duplica processo
- (3) wait()
 - a) Sincronização
 - b) Substitui o processo
 - c) Espera por filho
 - d) Duplica processo
- (4) pthread_mutex_lock()
 - a) Sincronização
 - b) Substitui o processo
 - c) Espera por filho
 - d) Duplica processo

17. Associe:

- (1) sem_open
 - a) decremento
 - b) remoção do nome
 - c) criação/abertura
 - d) incremento
- (2) sem_wait
 - a) decremento
 - b) remoção do nome
 - c) criação/abertura
 - d) incremento
- (3) sem_post
 - a) decremento
 - b) remoção do nome
 - c) criação/abertura
 - d) incremento
- (4) sem_unlink
 - a) decremento
 - b) remoção do nome
 - c) criação/abertura
 - d) incremento

Completar Código

18. Complete para abrir um ficheiro texto:

```
FILE *f = fopen("dados.txt", "_____");
```

19. Complete código para criar processo:

```
pid_t p = _____();  
if (p == 0) { printf("Filho"); }
```

20. Complete código para criar thread:

```
pthread_t t;  
pthread_create(&t, NULL, _____, NULL);
```