

MC514—Sistemas Operacionais:
Teoria e Prática

Profa. Islene Calciolari Garcia

Prova 2

18 de junho de 2009

Nome:

RA:

Questão	Nota
1	
2	
3	
4	
5	
6(a)	
6(b)	
Total	

Instruções: Você pode fazer a prova a lápis (desde que o resultado final seja legível :-). Não é permitida consulta a qualquer material manuscrito ou impresso. Em caso de fraude, todos os envolvidos receberão nota zero.

- (2.0) Enumere e explique resumidamente as principais responsabilidades de um sistema operacional.
- (0.5) Qual é a diferença entre um processo CPU-bound e um processo I/O-bound?
 - (0.5) É sempre possível determinar se um processo é CPU-bound ou I/O-bound por meio de uma análise do seu código?
 - (0.5) E em tempo de execução?
 - (0.5) Processos I/O bound merecem prioridade mais alta nos algoritmos de escalonamento? Justifique todas as respostas.
- (1.0) É possível criar um link simbólico para um arquivo que não existe? E um hard link? Justifique sua resposta.
- (1.0) O que são *device drivers*? Como os device drivers podem ser acoplados a um sistema operacional?
- (1.5) Desenhe um diagrama e explique como funciona a paginação em vários níveis. Quando esta abordagem é interessante?
- Os itens desta questão (no verso) consideram uma política de gerência de memória baseada em **paginação**.

- (a) (1.0) Descreva razões pelas quais a segunda função pode ter desempenho inferior à primeira.

```
char mat[N][PAGESIZE];

void preenche_rapido(char c) {
    int i, j;
    for (i = 0; i < N; i++)
        for (j = 0; j < PAGESIZE; j++)
            mat[i][j] = c;
}

void preenche_lento(char c) {
    int i, j;
    for (j = 0; j < PAGESIZE; j++)
        for (i = 0; i < N; i++)
            mat[i][j] = c;
}
```

- (b) (1.5) Analise o código abaixo e descreva o erro que irá ocorrer. Qual é a saída esperada do programa?

```
#include <stdio.h>

void f() {
    int v[5];
    int i;
    printf("Função f!\n");
    for (i = 0; i < 100; i++)
        v[i] = i;
    f();
}

int main() {
    f();
    return 0;
}
```

Boa prova!