

SISTEMAS OPERACIONAIS

Prof^a. Roberta Lima Gomes (soufes@gmail.com)

2^a. Lista de Exercícios – Parte 1

Data de Entrega: não há. O objetivo da lista é ajudar no estudo individual dos alunos.

IPC

1. Qual a diferença entre pipes e named pipes (fifos)?
2. Sobre o código a seguir, responda:

```
#define SIZE 6
#define READ 0
#define WRITE 1
main()
{
    pid_t pid1, pid2, pid3, pid4; int status;
    int fd[2]; char buffer[SIZE+1];
    struct rusage usage;
    pipe(fd);
    if ((pid1=fork())==0) {           // child 1
        while(1) {
            read(fd[READ], buffer, SIZE);
            buffer[SIZE]='\0';
            write(fd[WRITE], "tomato", SIZE);
        }
    } else if ((pid2=fork())==0) { // child 2
        while(1) {
            read(fd[READ], buffer, 6);
            buffer[6]='\0';
            write(fd[WRITE], "turnip", 6);
        }
    } else { // parent
        write(fd[WRITE], "potato", 6);
        fprintf(stderr, "Parent: I wrote a potato!\n", buffer);
        sleep(3);
        read(fd[READ], buffer, 6); buffer[6]='\0';
        fprintf(stderr, "Parent: I got back a %s!\n", buffer);
        kill(pid1, SIGINT); pid3 = wait(&status, 0, &usage);
        kill(pid2, SIGINT); pid4 = wait(&status, 0, &usage);
    }
}
```

- a) O que este código faz?
- b) O que acontece se omitirmos o comando write do pai?
- c) É possível prever qual será a saída do programa?
- d) O que acontece se omitirmos o comando write do um dos filhos?

3. Como são definidas e usadas áreas de memórias compartilhadas entre processos no UNIX.

4. Compare o uso de *pipes* com o uso da memória compartilhada como um mecanismo para a comunicação de dados entre os processos em um único computador. Cite duas das vantagens de cada método. Quando cada um poderia ser preferível?
5. Considere um esquema de comunicação entre processos em que mailboxes são usadas. Projete um esquema que possibilite um processo esperar até que uma mailbox esteja vazia, ou explique porque tal esquema não pode existir.
6. Como você emularia o comportamento de um semáforo em um sistema distribuído usando troca de mensagens?

THREADS

7. Explique a diferença entre unidade de alocação de recursos e unidade de escalonamento?
8. Quais as vantagens e desvantagens do compartilhamento do espaço de endereçamento entre threads de um mesmo processo?
9. Quais as diferenças entre threads em nível de usuário e threads em nível de kernel? Em quais circunstâncias um tipo é melhor do que o outro?
10. Quais estruturas são utilizadas para a criação de uma thread de kernel? Como elas se diferenciam daquelas utilizadas para a criação de um processo?
11. Como o uso de threads pode melhorar o desempenho de aplicações paralelas em ambientes com múltiplos processadores?
12. Usando os mecanismos "mutex" e "variáveis condição" do Posix, implemente as operações P e V de um semáforo. Cada uma dessas operações deverá ser substituída por um código C com semântica similar. Lembre-se que estas operações devem ser atômicas. O valor inicial do semáforo é 1.

Dicas sobre o Posix:

```
pthread_mutex_t m = PTHREAD_MUTEX_INITIALIZER;
```

```
pthread_mutex_lock( &m );
```

```
pthread_mutex_unlock( &m );
```

```
pthread_cond_t vc = PTHREAD_COND_INITIALIZER;
```

```
pthread_cond_wait( &vc , &m );
```

```
pthread_cond_signal( &vc );
```

28. Em Java, tipicamente quando uma thread realiza um "wait()", isto é feito dentro de um loop "while". Explique por quê.