

1. (2,0 ponto) Quais as diferenças entre o multiprocessamento simétrico e assi-métrico. Explique as diferenças quanto a arquitetura.

<i>Simétrico</i>	<i>Assimétrico</i>
<i>Cada processador executa as tarefas no SO</i>	<i>Apenas o processador mestre executa as tarefas do SO</i>
<i>Processadores com mesma arquitetura</i>	<i>Arquiteturas iguais ou diferentes</i>
<i>Cada processador pode executar processos tarefa conforme uma fila pronta ou privada</i>	<i>O processador mestre define as tarefas aos escravos</i>
<i>Comunicação por memória compartilhada</i>	<i>Não há necessidade de se comunicar, pois são controlados pelo Mestre</i>
<i>Se um processador falhar a capacidade do sistema é reduzida</i>	<i>Se um processador falhar o mestre seleciona um escravo para substituir</i>
<i>Processadores precisam ser sincronizados por motivo de equilíbrio de carga. Evitar ociosidade.</i>	<i>Quem controla a estrutura de dados é o Mestre.</i>

2. (2,0 pontos) Explique a diferença entre uma chamada de sistema (syscall) e uma API.

<i>API</i>	<i>System Call</i>
<i>Um conjunto de protocolos, rotinas, funções que os programadores usam para desenvolver software para facilitar as interações entre sistemas distintos.</i>	<i>Uma maneira programática pela qual um programa de computador solicita um serviço do kernel do sistema operacional em que é executado.</i>
<i>Ajuda a trocar dados entre vários sistemas, dispositivos e aplicativos.</i>	<i>Permite que um programa acesse serviços do kernel do sistema operacional.</i>

3. (2,0 pontos) Digamos que você tivesse que implementar no Sistema Operacional a operação de copiar um arquivo origem para um destino. Qual seria a sequência de operações necessárias? Basta pensar como você implementaria isso em sua linguagem preferida, digamos que o nome desse comando seja `cpy nomeOrigem nomeDestino`. Não precisa descrever os comandos da sua linguagem, mas o algoritmo.

```
#include <stdio.h>
#include <string.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>

#define TAM_BUFFER 1000

int main(int argc, char *argv[]){
    int descArquivoOrigem, descArquivoDestino, nBytesLidos,
nBytesGravados;
```

```

char Buffer[TAM_BUFFER];

if(argc!=3){
    printf("São necessário 3 parâmetros: cpy arqOrigem arqDestino\
n");
    exit(1);
}
    if((descArquivoOrigem=open(argv[1],O_RDONLY))<0){ //se tudo
correu bem o descritor será maior que zero, se -1 deu erro
    printf("Erro na abertura do arquivo origem %s\n", argv[1]);
    fprintf(stderr, "Erro: %s\n", strerror(errno));
    exit(1);
}
//else printf("Abertura Origem ok..\n");

    descArquivoDestino=open(argv[2],O_WRONLY);//tenta abrir, se
existe abre
    if (descArquivoDestino > 0){ //SUGESTAO:Criar um menu
perguntando se deseja sobrepor o existente
    printf("Arquivo destino já existente\n");
    close(descArquivoDestino);
    exit(1);
}
else{

        if((descArquivoDestino=open(argv[2],O_CREAT | O_WRONLY,
0660))<0){ // 660 - rw- rw- ---
            printf("Erro na abertura do arquivo destino %s\n",
argv[2]);
            close(descArquivoOrigem);
            fprintf(stderr, "Erro: %s\n", strerror(errno));
            exit(1);
        }
        //else printf("Abertura Destino ok..\n");

}

//else
    while((nBytesLidos=read(descArquivoOrigem, Buffer,
TAM_BUFFER))>0){
        nBytesGravados=write(descArquivoDestino, Buffer,
nBytesLidos);
        //printf("Bytes lidos: %d, Bytes gravados: %d\
n",nBytesLidos,nBytesGravados);
        if(nBytesLidos!=nBytesGravados){
            printf("Erro na gravação do arquivo %s\n",argv[2]);
            fprintf(stderr, "Erro: %s\n", strerror(errno));

```

```
    close(descArquivoDestino);
    close(descArquivoOrigem);
    exit(1);
}
} //while
printf("Copia ocorrida com sucesso...\n");
close(descArquivoDestino);
close(descArquivoOrigem);
return 0;
}
```

4. (2,0 pontos) Descreva o que é um Sistema Operacional. Esquematize se achar que explica melhor.

Podê ser considerado como um programa que permite a iteração do usuário com o computador.

Permite o uso do computador de maneira eficiente e fácil, permite o uso de várias aplicações afins.

5. (2,0 pontos) Explique o conceito de máquina virtual. Qual a vantagem de utilizar este conceito?

Uma máquina virtual permite que se trabalhe com diferentes sistemas operacionais ou arquiteturas em sistemas e arquiteturas diferentes. Para isso a máquina virtual cria uma camada que permite ao software ou SO usar os recursos que necessita sem interagir com o hardware real.