**3.1 Kernel**

O kernel é um gerenciador de recursos, caso o recurso que está sendo gerenciado for um processo, uma memória ou um dispositivo de hardware, ele gerencia e intermedeia o acesso entre os vários usuários concorrentes. É um conjunto de rotinas que oferecem serviços aos usuários de sistema. As principais funções são: tratamento de interrupções, criação e eliminação de processos, sincronização e comunicação entre processos, escalonamento e controle dos processos, gerência de memória, gerência do sistema de arquivos, operações de entrada e saída, contabilização e segurança do sistema.

**3.1.1 Chamadas de sistema**

Também conhecidas como System Calls ou Syscalls, são primitivas através das quais os programas requisitam algum serviço do kernel do sistema operacional. Estes serviços podem acessar o disco rígido, criar e executar um novo processo. Sendo chamadas de sistema, são meios de comunicação entre o programa, processo e o sistema operacional. As chamadas de sistema são responsáveis de ser a interface entre o programa em execução e o kernel. A comunicação entre o kernel e o hardware é realizada pelos drivers através dos subsistemas de cada dispositivo.

**3.1.2 Kernel Monolítico**

É uma arquitetura de kernel onde é executado no espaço de kernel no modo de supervisão. Ou seja, é um kernel que possui todos os códigos de suporte necessários. O kernel define uma camada de alto nível de abstração sobre o hardware do computador, com um conjunto de primitivas ou chamadas de sistema para implementar os serviços do sistema operacional como gerenciamento de processos, concorrência e gestão de memória em um ou mais módulos.

**3.1.4 Kernel Monolítico X Microkernel**

O kernel monolítico é um grande processo único executado inteiramente em um único espaço de endereço. É um único arquivo binário estático. Todos os serviços do kernel existem e são executados no espaço de endereço do kernel. O kernel pode invocar funções diretamente. Exemplos de SOs baseados em kernel monolítico: Unix, Linux.

Nos microkernels, o kernel é dividido em processos separados, conhecidos como servidores. Alguns dos servidores são executados no espaço do kernel e alguns são executados no espaço do usuário. Todos os servidores são mantidos separados e executados em espaços de endereço diferentes. Os servidores invocam "serviços" uns dos outros enviando mensagens via IPC (comunicação entre processos). Essa separação tem a vantagem de que, se um servidor falhar, outros servidores ainda poderão funcionar com eficiência. Exemplos de sistemas operacionais baseados em microkernel: Mac OS X e Windows NT.