

Nome: <a href="#">Pedro Gabriel Garcia Ribeiro Balestra</a>	Matricula: <a href="#">1551</a>
Curso: <a href="#">GEC</a>	Período: <a href="#">P8</a>
	Matéria: <a href="#">C012</a>

## Cap.2 – Estrutura do Sistema Operacional

1. User Interface (UI) basicamente é tudo aquilo que leva ao usuário algum tipo de interação com a plataforma usada, podendo ser um botão, menu, som ... Temos como exemplo de interação o CLI (Command Line Interface) onde o usuário pode interagir com o SO através de linhas de comando e temos também o GUI (Graphical User Interface) onde temos interfaces mais amigáveis para o usuário que utilizam o curso do mouse e ícones para a interação.
2. São chamadas realizadas pelo sistema, onde são responsáveis pela alocação de recursos, arquivos de sistema, comunicação, proteção dentre muitas outras operações realizadas no SO, sendo implementadas por classes, conectas a APIs.

3.

<b>System Calls</b>	<b>Windows</b>	<b>Linux</b>
<b>Controle de Processos</b>	ExitProcess()	exit()
<b>Manipulação de Arquivos</b>	ReadFile()	read()
<b>Manipulação de Dispositivos</b>	Writeconsole()	write()
<b>Manipulação de Informações</b>	GetCurrentProcessID()	getpid()
<b>Comunicação</b>	CreatePipe()	pipe()
<b>Proteção</b>	SetFileSecurity()	chmod()

4. Para qual dispositivo o SO será feito, PC ou dispositivo móvel, se será usado por somente um usuário ou por vários, irá executar várias tarefas ao mesmo tempo ou em tempo real, dentre outros...
5. O System Generation (SysGen) será rodado durante a instalação do SO para poder entender como o usuário deseja deixar seu SO, como a quantidade de partições HD/SSD, dispositivos I/O presentes entre outros
6. É uma sequência de ações realizadas pelo SO afim de encontrar erros, Quando encontrado um erro o SO fará um pesquisa para encontrar a fonte do erro, e irá tentar repara-lo, para evitar que a famosa “Tela Azul”, o sistema gera logs que armazenam as memory dumps, que são “fotografias” da RAM no momento do erro, que com eles o SO consegue voltar no ultimo processo feito antes do erro.

7. Programas de Sistema, são os programas que estão presentes na instalação do SO, onde muitos deles são importantes para o funcionamento do SO, como os loaders, linkers etc. Muitos podem ser baixados posteriormente para o auxílio de atividades, como o suporte para linguagem de programação, compiladores, interpretadores etc.
8. Arquitetura simples -> Maior simplicidade e desempenho, porém tem uma dificuldade de implementação e manutenção.  
Arquitetura em camadas -> Tem modularidade, facilidade de depuração e modificação, porém tem um menor desempenho  
Arquitetura de MicroKernel -> Possui uma facilidade de extensão do SO, porém tem um desempenho comprometido devido ao overhead(sobrecarga) de funções do sistema  
Arquitetura em Módulos -> Os módulos são chamados conforme a necessidade, onde cada modulo pode chamar serviços de outros módulos e se comunicar, porém tem uma maior complexidade de desenvolvimento.
9. São ambientes de SOs, que rodam sobre o SO da máquina, ficando acima de uma camada de software, podendo rodar outro SO na VM, onde ele fará comunicações diretamente com o hardware. A VM pode ocorrer de três maneiras:  
Virtualização total -> todo o hardware fala com o SO instalado e a VM roda em cima de uma camada de software.  
Paravirtualização -> O SO instalado sabe que está sendo virtualizado e dessa forma é modificado para tirar o maior proveito do hardware, dessa forma devemos ter um hardware mais potente. Podendo rodar mais de um SO por vez.  
Emulação -> É o processo de emular um hardware em SO, temos com exemplo, os emuladores de jogos no computador.