Insper

Tecnologias Hacker

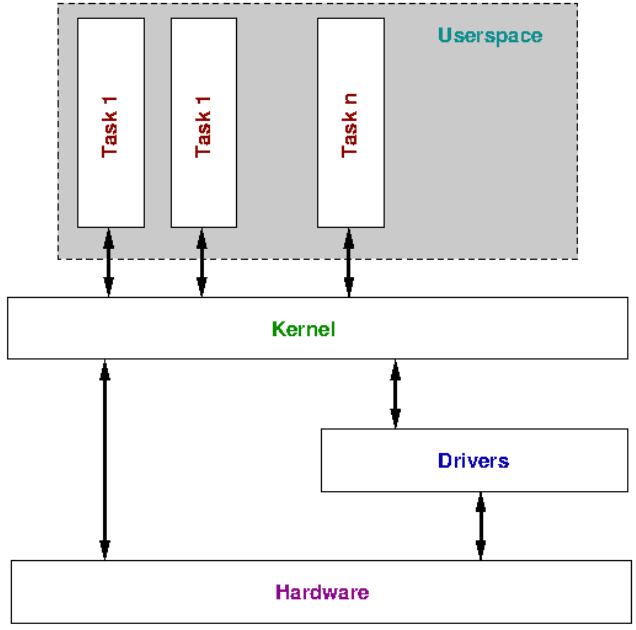
Aula Sistemas Operacionais

Objetivos da aula

Retomar conceitos fundamentais na administração de um sistema operacional GNU/Linux, abordando pontos importantes para a proteção e exploração de sistemas.

Processos e Daemons

Organização do sistema operacional



The organisation of the kernel. Processes the kernel is running live in *userspace*, and the kernel talks both directly to hardware and through *drivers*.

https://www.bottomupcs.com/organisation_of_os.xhtml

Processos

Cada sistema tem um objetivo específico que deseja alcançar. Esse objetivo poderia ser fornecer um site para visitantes anônimos em todo o mundo. Para permitir isso, deve haver algo que escute as solicitações individuais do site, processe-as e, finalmente, envie de volta a página do site relacionado. Chamamos isso de processo e consiste em código de máquina. Estas são instruções individuais sobre o que o sistema deve fazer. Essas instruções incluem a leitura de uma imagem do disco rígido, o envio de dados pela interface de rede ou a gravação de uma mensagem de erro em um arquivo de log.

Listando os processos

pstree

```
avelino@matrix:/proc/22$ pstree
systemd─┬─ModemManager─┬─{gdbus}
        │              └─{gmain}
        ├─NetworkManager─┬─dhclient
        │                ├─{gdbus}
        │                └─{gmain}
        ├─accounts-daemon─┬─{gdbus}
        │                 └─{gmain}
        ├─anydesk─┬─{gdbus}
        │         ├─{gmain}
        │         └─{proc}
        ├─avahi-daemon───avahi-daemon
        ├─bluetoothd
        ├─chromium─┬─chrome-gnome-sh─┬─{gdbus}
        │          │                 └─{gmain}
        │          ├─chrome-sandbox───chromium───chromium─┬─13*[chromium─┬─{Chrome_ChildIOT}]
        │          │                                      │              ├─3*[{CompositorTileW}]]
        │          │                                      │              ├─{Compositor}]
        │          │                                      │              ├─{Font_Proxy_Thre}]
        │          │                                      │              ├─{GpuMemoryThread}]
        │          │                                      │              ├─2*[{TaskSchedulerFo}]]
        │          │                                      │              └─{TaskSchedulerSe}]
        │          │                                      └─24*[chromium─┬─{Chrome_ChildIOT}]
        │          │                                                     ├─3*[{CompositorTileW}]]
        │          │                                                     ├─{Compositor}]
        │          │                                                     ├─{Font_Proxy_Thre}]
        │          │                                                     ├─{GpuMemoryThread}]
        │          │                                                     ├─{MemoryInfra}]
```

ps e top

- são processos no nível do usuário (executamos por meio do shell -nível de usuário)
- eles exibem informações sobre processos que exigem uma visão de todo o sistema.
- No caso de top, essas informações são atualizadas dinamicamente, sendo assim, muitas estruturas no nível do SO precisam expor informações aos programas do usuário.

/proc

Como os sistemas fornecem esse tipo de informação aos processos no nível do usuário?

- Log de eventos – armazenados em disco e sobrecarga para criar arquivos de log.
- Acesso direto à memória do kernel - requer que os programadores de nível de usuário tenham conhecimento das estruturas de dados do kernel.

/proc

Os sistemas Linux optam por fornecer essas informações por meio de uma estrutura hierárquica semelhante a um arquivo, chamada de “sistema de arquivos virtual” /proc.

Este sistema permite que programas no nível do usuário acessem informações sobre processos e outras informações do sistema de maneira conveniente e padronizada.

Daemons

Alguns processos têm o objetivo de serem executados por um longo tempo no sistema em segundo plano.

Isso pode ser para atender a solicitações como verificar um e-mail recebido ou enviar uma página de um site.

Esses processos são chamados daemons. Além da duração, outra grande diferença é que os daemons não precisam de interação com o terminal.

Daemons

Normalmente não enviam nenhuma informação por meio da saída padrão e sim realizam seus registros em arquivos de log.

Geralmente são iniciados diretamente após o início do sistema operacional. A maioria tem um 'd' no final do nome do processo, para indicar que eles são um processo daemon.

Um daemon é sempre um processo, mas nem todos os processos são um daemon

Normalmente, o termo 'serviço' era usado em sistemas Windows. Com a introdução do systemd, esse termo agora é mais aplicável também ao Linux.

Um serviço é uma combinação de recursos para fornecer alguma funcionalidade. Por exemplo, um serviço SSH, que consiste em executar o daemon relacionado e quaisquer dependências como rede.

```
root@matrix:/proc/23127# ps -ef
UID          PID  PPID  C STIME TTY          TIME CMD
root           1     0  0 ago06 ?        00:00:02 /sbin/init
root           2     0  0 ago06 ?        00:00:00 [kthreadd]
root           3     2  0 ago06 ?        00:00:00 [ksoftirqd/0]
root           5     2  0 ago06 ?        00:00:00 [kworker/0:0H]
root           7     2  0 ago06 ?        00:00:32 [rcu_sched]
root           8     2  0 ago06 ?        00:00:00 [rcu_bh]
root           9     2  0 ago06 ?        00:00:02 [migration/0]
root          10     2  0 ago06 ?        00:00:00 [lru-add-drain]
root          11     2  0 ago06 ?        00:00:00 [watchdog/0]
root          12     2  0 ago06 ?        00:00:00 [cpuhp/0]
root          13     2  0 ago06 ?        00:00:00 [cpuhp/1]
root          14     2  0 ago06 ?        00:00:00 [watchdog/1]
root          15     2  0 ago06 ?        00:00:00 [migration/1]
root          16     2  0 ago06 ?        00:00:00 [ksoftirqd/1]
root          18     2  0 ago06 ?        00:00:00 [kworker/1:0H]
root          19     2  0 ago06 ?        00:00:00 [cpuhp/2]
root          20     2  0 ago06 ?        00:00:00 [watchdog/2]
root          21     2  0 ago06 ?        00:00:00 [migration/2]
root          22     2  0 ago06 ?        00:00:00 [ksoftirqd/2]
root          24     2  0 ago06 ?        00:00:00 [kworker/2:0H]
```

Shadow

Identificador 1 hash senha e algoritmo ($6)

$1 = Algoritmo de hash MD5.
$2 = Algoritmo de hash Blowfish.
$2a= Algoritmo de hash eksblowfish.
$5 = Algoritmo de hash SHA-256.
$6 = Algoritmo de hash SHA-512.

Identificador 2 > Última alteração de senha (última alteração) : dias desde 1º de janeiro de 1970 em que a senha foi alterada pela última vez.

Identificador 3 > Mínimo : O número mínimo de dias necessários entre as alterações de senha, ou seja, o número de dias restantes para que o usuário possa alterar sua senha.

Identificador 4 > Máximo : o número máximo de dias em que a senha é válida (depois que o usuário é forçado a alterar sua senha).

Identificador 5 > Aviso : o número de dias antes da senha expirar, o usuário é avisado de que sua senha deve ser alterada.