

Sistemas Operacionais

Conceitos

Felipe Augusto Lima Reis
felipe.reis@ifmg.edu.br



Sumário

- 1 Processos
- 2 Memória
- 3 Arquivos
- 4 Cache
- 5 I/O
- 6 Proteção e Seg.
- 7 System Calls

PROCESSOS

Processos

- “Processo é basicamente um programa em execução” [Tanenbaum and Bos, 2014];
 - “O programa é uma entidade passiva e por si só não é um processo” [Silberschatz et al., 2012]
 - O programa somente se torna um processo quando é executado;

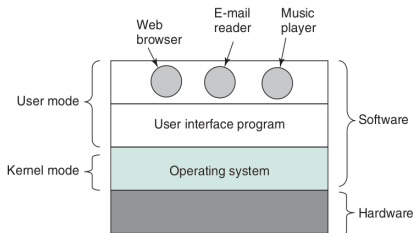
Processos



- Um processo precisa de recursos, como tempo de CPU, memória, acesso a dispositivos I/O
 - Os recursos podem ser alocados quando o processo é criado ou durante a execução do programa;
- Um processo é a unidade de trabalho em um sistema;
- Um sistema consiste em uma coleção de processos
- Um processo é fundamentalmente um contêiner que contém todas as informações necessárias para executar um programa [Tanenbaum and Bos, 2014] [Silberschatz et al., 2012].

Processos

- Processos podem ser divididos em dois tipos:
 - Processos do usuário (*user mode*): executam código do usuário, com um conjunto restrito de instruções;
 - Processos do SO (*kernel mode*): executam códigos do sistema operacional¹ [Tanenbaum and Bos, 2014].



Processos de um sistema operacional.

Fonte: [Tanenbaum and Bos, 2014]

¹ Provê acesso completo a todo o hardware e permite execução de qualquer instrução na máquina.

Processos

- Cada processo possui um **espaço de endereçamento**
 - Esse espaço corresponde a uma lista de locais em memória (de 0 até um valor máximo) onde o processo pode ler e escrever informações;
- O espaço de endereçamento contém:
 - O programa executável (ou em execução);
 - Os dados do programa;
 - Pilha (área especial na memória que armazena as variáveis temporárias criadas por uma função)
[Tanenbaum and Bos, 2014].

Processos

- Cada processo é identificado por um número único no Sistema Operacional, denominado **Process ID** ou **PID**²;
- Cada processo contém uma lista de **recursos necessários**:
 - Registradores (incluindo registradores especiais);
 - Lista de arquivos abertos, usados pelo processo;
 - Processos relacionados;
 - Qualquer outra informação extra necessária à execução do processo [Tanenbaum and Bos, 2014].

²Identificador do processo

GERÊNCIA DE PROCESSOS

Gerência de Processos

- A **gerência de processos** do sistema operacional é responsável pelas seguintes atividades:
 - Agendamento de processos e threads de CPU³;
 - Criar e remover processos de sistemas e de usuários;
 - Suspende e retomar a execução de processos;
 - Prover mecanismos para sincronização;
 - Prover mecanismos para comunicação entre processos [Silberschatz et al., 2012].

³Thread é uma forma de um processo dividir a si mesmo em duas ou mais tarefas, que podem ser executadas concorrentemente.

Suspensão de Processos

- A suspensão de processos é uma atividade frequente durante o uso de sistemas operacionais;
- Suponha que vários processos tenham sido iniciados e são utilizados pelo usuário
 - De tempos em tempos, o sistema operacional interrompe a execução de um processo para execução de outro;
 - Esse procedimento é realizado constantemente no sistema operacional, muitas vezes por segundo;
 - Quando o processo é suspenso, o mesmo deve ser salvo, para que a execução seja iniciada no mesmo ponto em que o processo foi interrompido [Tanenbaum and Bos, 2014].

Suspensão de Processos

- Em muitos sistemas operacionais, as informações dos processos são armazenadas em uma **Tabela de Processos**
 - Quando um processo é suspenso, seus dados são escritos na tabela de processos;
 - Quando um processo é reiniciado, seus dados são lidos da tabela de processos;
- Um processo suspenso consiste em seu espaço de endereços, sua tabela individual de processos, conteúdos dos registradores e informações necessárias para reinício [Tanenbaum and Bos, 2014].

Criação e Finalização de Processos

- Processos podem ser criados com a inicialização de programas⁴ pelo usuário ou pelo sistema operacional;
- Processos também podem criar **processos filhos**
 - Processos filhos, criados por um processo pai (principal), gerando uma árvore de processos;
 - Ex.: Google Chrome cria um processo pai e processos filhos para cada aba.
 - Caso seja encerrado somente um processo filho, neste exemplo, Chrome indicaria falha em apenas na aba encerrada;
 - Caso seja encerrada a árvore de processos, toda a aplicação é finalizada.

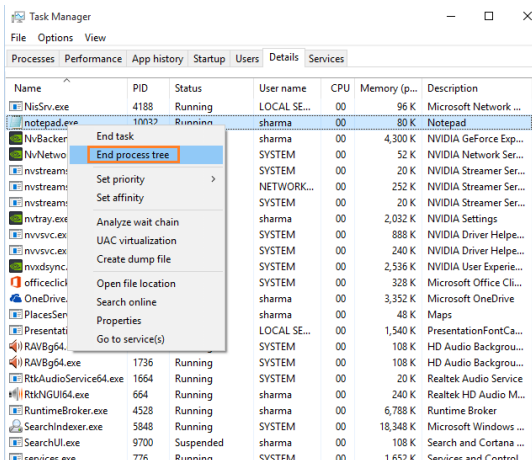
⁴Conforme definição de processo, no Slide 4.

Criação e Finalização de Processos

- Processos podem ser criados com a inicialização de programas⁴ pelo usuário ou pelo sistema operacional;
- Processos também podem criar **processos filhos**
 - Processos filhos, criados por um processo pai (principal), gerando uma árvore de processos;
 - Ex.: Google Chrome cria um processo pai e processos filhos para cada aba.
 - Caso seja encerrado somente um processo filho, neste exemplo, Chrome indicaria falha em apenas na aba encerrada;
 - Caso seja encerrada a árvore de processos, toda a aplicação é finalizada.

⁴Conforme definição de processo, no Slide 4.

Criação e Finalização de Processos



Fechar árvore de processos no Windows 10.

Fonte: [The Geek Page, 2016]

Comunicação entre Processos

- Processos relacionados, ou filhos, podem cooperar para conclusão de uma determinada tarefa;
- Processos podem ter necessidade de se comunicar, a fim de sincronizar as atividades que serão executadas;
 - Esse mecanismo de comunicação é denominado **interprocess communication** (comunicação entre processos) [Tanenbaum and Bos, 2014].

MEMÓRIA PRINCIPAL

Memória Principal



- A memória principal é um repositório de dados compartilhados que pode ser acessado rapidamente pela CPU de dispositivos de I/O;
- O processador pode ler e escrever instruções / dados na memória
 - O processador lê instruções da memória principal durante o ciclo de busca de instrução;
 - O processador lê e escreve dados durante o ciclo de busca de dados [Silberschatz et al., 2012].

Espaço de Endereçamento

- Um espaço de endereçamento corresponde a uma lista de locais em memória, iniciando no valor 0 até o máximo de memória disponível [Tanenbaum and Bos, 2014];
 - Nesse local podem ser lidos e escritos valores pelo processo;
 - Contém o processo, seus dados e a pilha de execução;
 - Utilizam 32 ou 64 bits para endereçamento
 - 32 bits (x86): possível endereçar até 2^{32} bits (aprox. 4GB RAM);
 - 64 bits (x86-64): possível endereçar até 2^{64} bits (aprox. 16EB RAM);

Importante: O Windows “reconhece” no máximo 3GB em sistemas 32 bits e 16TB (2^{44}) em sistemas de 64 bits.

Espaço de Endereçamento

- Um espaço de endereçamento corresponde a uma lista de locais em memória, iniciando no valor 0 até o máximo de memória disponível [Tanenbaum and Bos, 2014];
 - Nesse local podem ser lidos e escritos valores pelo processo;
 - Contém o processo, seus dados e a pilha de execução;
 - Utilizam 32 ou 64 bits para endereçamento
 - 32 bits (x86): possível endereçar até 2^{32} bits (aprox. 4GB RAM);
 - 64 bits (x86-64): possível endereçar até 2^{64} bits (aprox. 16EB RAM);

Importante: O Windows “reconhece” no máximo 3GB em sistemas 32 bits e 16TB (2^{44}) em sistemas de 64 bits.

Arquitetura x86 e x86-64

- A arquitetura x86 (32 bits) tem origem na família de processadores 80x86, baseados na arquitetura do Intel 8086
 - O nome da arquitetura deriva do sufixo “86”, utilizado para identificação dos primeiros processadores;
- Os primeiros processadores x86 - Intel 8086 (1978), 80186 (1982) e 80286 (1982) - possuíam 16 bits de endereçamento;
- A partir do Intel 80386 (1985), a arquitetura passou a contar com 32 bits para endereçamento;
- Processadores da família Intel Pentium, AMD K6 e Athlon também possuíam 32 bits.

Arquitetura x86 e x86-64

- O processador **Itanium**, da Intel, lançado em 2001, era puramente 64 bits, não executando instruções 32 bits;
- Em 2003, a AMD lançou a família de processadores AMD K8, capaz de executar instruções de 32 e 64 bits;
 - Essa arquitetura foi chamada de x86-64 ou AMD64;
 - Processadores foram chamados comercialmente de **Opteron64** (para servidores) e **Athlon64** (para desktops);
- Atualmente, tanto processadores Intel quanto AMD aceitam instruções de 32 e 64 bits.

Gerência de Memória

- A **gerência de memória** do sistema operacional é responsável pelas seguintes atividades:
 - Manter informações sobre quais partes da memória estão sendo utilizadas e por quem;
 - Decidir quais processos (ou partes dos processos) devem ser movidas para dentro/fora da memória;
 - Alocar e desalocar espaços na memória assim que for solicitado [Silberschatz et al., 2012].

ARQUIVOS

Arquivos

- O sistema operacional deve abstrair propriedades físicas da gravação de dados, definindo uma unidade de lógica de armazenamento, o **arquivo** [Silberschatz et al., 2012];
- Os arquivos são compostos por um nome (obrigatório) e uma extensão (opcional)
 - Nome do arquivo é utilizado para identificação textual da informação (para humanos);
 - Recomenda-se que nome do arquivo não possua espaços nem acentuação;
 - A extensão indica, em alguns sistemas, como o Windows, o tipo do arquivo.

Diretórios



- Para organização, sistemas operacionais possuem o conceito de **diretório**, que permite o agrupamento de arquivos;
 - Diretórios apresentam, em geral, o conceito de hierarquia;
 - A hierarquia de diretórios pode ser representada como uma **árvore de diretórios**;
 - O diretório mais alto na hierarquia é comumente chamado de **diretório raiz** (*root directory*).
 - Para acessar arquivos na árvore de diretórios, é utilizado um **caminho** (*path*);
 - O caminho é composto por todos os diretórios/subdiretórios até o arquivo, iniciando a nomenclatura no diretório raiz;

Diretórios



Árvore de diretórios.

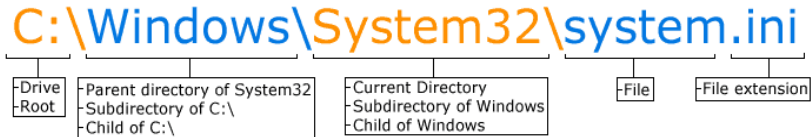
Fonte: [Wikipedia Contributors, 2019]

Arquivos e Diretórios

- Ambiente Windows

- Diretórios iniciam com a letra correspondente ao dispositivo de armazenamento (HD / SSD / DVD /etc);
- Por questões históricas, normalmente, o disco principal utiliza a letra C (C:);
 - Unidades de disco “flexíveis” (*floppy disks* - disquetes) utilizavam as letras A e B;
 - Unidades de disco rígido (*hard disks*) utilizavam letras C, D, E, F, etc.;
- Cada diretório/subdiretório dentro de um caminho é separado pelo caractere de barra invertida (\);
 - Ex.: C:\Windows\System32\system.ini

Arquivos e Diretórios



Caminho de um arquivo (Windows).

Fonte: [Computer Hope, 2018]

Arquivos e Diretórios



- Ambiente Linux

- Diretórios em Linux iniciam-se com barra comum (/);
- O diretório raiz (*root*) inicia-se com uma barra simples (/);
- Não existe unidade de disco, como no Windows;
 - Os demais discos, caso estejam montados (disponíveis para uso), geralmente estão localizados na pasta */media/*;
 - Discos não montados (indisponíveis para uso) são encontrados, na maioria das distribuições, na pasta */dev/*;
 - A montagem dos discos pode ser feita pela interface do sistema ou linha de comando (será abordado no futuro);
 - Algumas distribuições Linux, como o Ubuntu, realizam montagem automática de dispositivos como *pen-drives*;

Arquivos e Diretórios

```
/home/hope/public_html/
```

Caminho de um arquivo (Linux).

Fonte: [Computer Hope, 2018]

- A estrutura de diretório possui o seguinte significado:
 - `/home`: pasta padrão para criação de arquivos dos usuários do sistema;
 - `/hope`: nome do usuário;
 - `/public_html`: pasta de arquivos públicos de um servidor web (ou FTP).

Arquivos

- A **gerência de arquivos e armazenamento** do sistema operacional é responsável pelas seguintes atividades:
 - Criar e excluir arquivos e diretórios;
 - Gerar primitivas para manipulação de arquivos por programas;
 - Realizar a coordenação de espaços livres em disco;
 - Criar cópias de arquivos em mídias secundárias [Silberschatz et al., 2012].

CACHE

Cache

- **Cache** é um dispositivo de acesso rápido, que serve de intermediário entre um processo e um dispositivo de armazenamento
 - As informações são mantidas em algum sistema de armazenamento (como a memória principal);
 - À medida em que são usadas, elas são copiadas para sistemas de armazenamento mais rápidos - chamados de cache - temporariamente.

Cache

- Quando precisamos de uma determinada informação, primeiro verificamos se ela está no cache
 - Se ela estiver no cache, utilizamos essa informação;
 - Caso contrário, recuperamos a informação no sistema de armazenamento convencional e colocamos uma cópia no cache, supondo que precisaremos novamente dessa informação [Silberschatz et al., 2012].

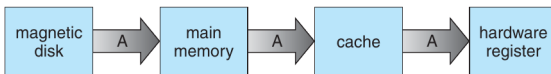
Cache

- Responsabilidade pelo gerenciamento de cache:
 - Algumas políticas de cache são implementadas pelo hardware ou pelos compiladores (registradores e cache de CPU);
 - Outras políticas são implementadas pelos sistemas operacionais (memórias principais e secundárias);
- Devido às limitações de tamanho de cache, as políticas de **gerenciamento de cache** podem influenciar consideravelmente no desempenho do sistema [Silberschatz et al., 2012].

Existem diversas técnicas para substituição de informação no cache, chamadas de políticas de substituição cache. Essas políticas, no entanto, não serão detalhadas nesta seção.

Cache

- Alguns dispositivos podem ser considerados “caches” de outros
 - A memória primária (RAM) pode ser considerada um “cache” da memória secundária (HD, SSD);
 - A existência do cache dá origem à hierarquia de armazenamento;
 - Um dado armazenado, por exemplo, em um HD deve percorrer toda a hierarquia de memória até pode ser utilizado pelo processador.



Caminho de um inteiro “A”, do HD para um registrador.

Fonte: [Silberschatz et al., 2012]

Cache

- Devido ao desempenho do componentes computacionais, justifica-se a utilização de diferentes níveis de cache
 - Ressalta-se que componentes mais próximos à CPU tendem a ter um custo financeiro mais alto⁵.

Level	1	2	3	4	5
Name	registers	cache	main memory	solid state disk	magnetic disk
Typical size	< 1 KB	< 16MB	< 64GB	< 1 TB	< 10 TB
Implementation technology	custom memory with multiple ports CMOS	on-chip or off-chip CMOS SRAM	CMOS SRAM	flash memory	magnetic disk
Access time (ns)	0.25 - 0.5	0.5 - 25	80 - 250	25,000 - 50,000	5,000,000
Bandwidth (MB/sec)	20,000 - 100,000	5,000 - 10,000	1,000 - 5,000	500	20 - 150
Managed by	compiler	hardware	operating system	operating system	operating system
Backed by	cache	main memory	disk	disk	disk or tape

Desempenho de hardwares na hierarquia de armazenamento.

Fonte: [Silberschatz et al., 2012]

⁵ Alguns dispositivos também possuem como problema adicional sua volatilidade, ou seja, perdem informação armazenada se o fornecimento de energia for desligado.

Cache

- Devido ao mecanismo de cache, uma mesma informação pode existir em diferentes níveis da hierarquia de armazenamento;
 - Isso ocorre pois uma informação normalmente é duplicada nas diferentes hierarquias (*inclusive caches*)⁶;
 - O processador sempre acessa a informação que está mais próximo a ele na hierarquia;
 - Caso haja alteração de informação, todos os níveis de informação devem ser alterados para que exista **coerência de cache**.

⁶ Algumas arquiteturas permitem caches exclusivos, onde um dado está somente em um nível da hierarquia (ex.: caches L1 e L2 de processadores AMD Athlon).

Coerência de Cache

- Em sistemas monoprocessados, é relativamente simples a manutenção da **coerência de cache**;
- No entanto, em sistemas multiprocessados, a coerência de cache pode se tornar consideravelmente complexa
 - Uma cópia de uma informação pode existir em múltiplos caches;
 - Ao alterar uma informação em um cache, as demais cópias devem imediatamente serem alteradas nos demais níveis;
- Em sistemas distribuídos, a coerência de cache pode se tornar ainda mais complexa, devido à latência entre computadores [Silberschatz et al., 2012].

INPUT/OUTPUT

Input/Output

- Computadores possuem dispositivos que possibilitam a **entrada e saída de informações**;
- Os sistemas operacionais implementam **subsistemas para gerenciamento de dispositivos I/O**;
 - Os subsistemas podem implementar softwares genéricos ou dedicados a um determinado dispositivo;
 - Existe também a possibilidade do software comunicar com o sistema operacional por software proprietário
 - São escritos pelo construtores do hardware;
 - Chamados de **device drivers** (*drivers* de dispositivos);
 - Os **drivers** correspondem a interfaces de software para comunicação com o hardware;

Input/Output

- O subsistema de I/O consiste nos seguintes componentes:
 - Gerência de memória dos componentes, que incluem:
 - **Buffer**: região de memória utilizada para armazenar temporariamente dados que estão sendo movidos de um lugar para outro;
 - **Cache**: dispositivo rápido que serve como intermediário entre o processo e o dispositivo de armazenamento;
 - **Spool**: processo de transferência de dados, onde estes são colocados em uma área temporária para que programa pode acessá-lo; conceito muito comum quando se trata de impressão de documentos.
 - Interface genérica de driver de dispositivos;
 - Drivers para dispositivos específicos [Silberschatz et al., 2012].

Drivers de dispositivos

- *Drivers de dispositivos* podem ser nativos dos sistemas operacionais ou instalados pelo usuário;
 - *Drivers* nativos, muitas vezes, são implementados de forma genérica, o que permite que o mesmo driver funcione em diversos dispositivos;
 - *Drivers* específicos conhecem as peculiaridades do dispositivo e podem obter melhor desempenho;
 - *Drivers* genéricos podem não aproveitar adequadamente todos os recursos do hardware.

PROTEÇÃO E SEGURANÇA

Proteção



- **Proteção:** mecanismo para controle de acesso de processos ou usuários a recursos definidos pelo sistema computacional [Silberschatz et al., 2012];
 - Esses mecanismos devem garantir que arquivos, segmentos de memória e CPU possam ser operados apenas pelos processos que possuem a devida autorização;
 - A autorização para operação é dada pelo sistema operacional;
 - Caso um processo tente acessar um dispositivo não autorizado, este deve ser reconhecido e bloqueado pelo sistema operacional;

Proteção



- Mecanismos de proteção possuem as seguintes responsabilidades secundárias:
 - Melhorar a confiabilidade detectando erros nas interfaces entre os subsistemas de componentes;
 - Prevenir que um subsistema em mal funcionamento interrompa o funcionamento de outros componentes que trabalham de forma adequada;

Segurança

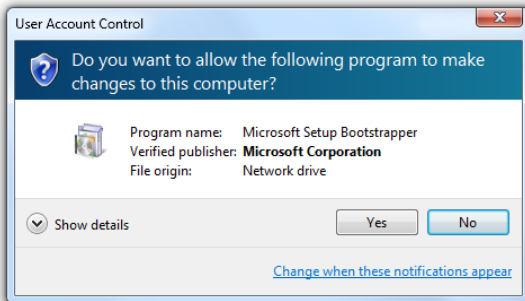
- **Segurança**: funcionalidade do sistema responsável pela defesa contra ataques internos e externos [Silberschatz et al., 2012];
 - **Ataques internos**: programas instalados ou executados no computador que tentam enviar dados para outros dispositivos, armazenar informações confidenciais, roubar dados, remover/modificar documentos/programas, etc.;
 - **Ataques externos**: ataques oriundos de outros computadores com objetivo de roubar/alterar documentos, interromper serviços (*denial-of-service*), roubo de identidade, etc.

Recursos de Proteção e Segurança



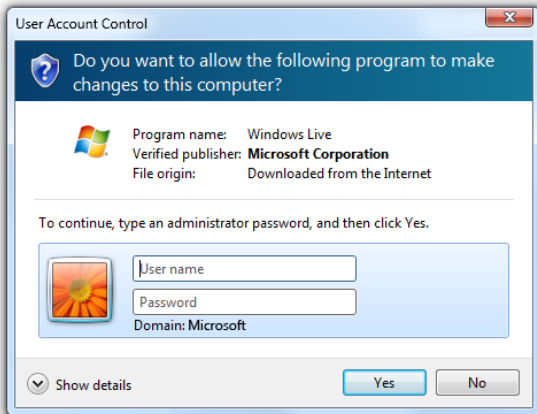
- Usuários e grupos:
 - **User ID** (Linux) ou **Security ID** (Windows) são números únicos correspondentes aos usuários do sistema operacional;
 - **Group identifier** (identificador de grupos) são nomes únicos que designam conjunto de usuários com acesso a um determinado recurso;
 - **Escalada de privilégios** correspondem a solicitação de confirmação do usuário, inserção de senha (ou usuário + senha) para que atividades possam ser executadas/realizadas;

Escalada de privilégios



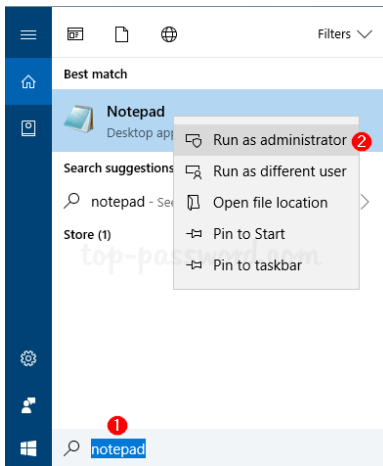
Escalada de privilégios (Windows).
Fonte: [Lucideus - Medium, 2018]

Escalada de privilégios



Escalada de privilégios (Windows).
Fonte: [Lucideus - Medium, 2018]

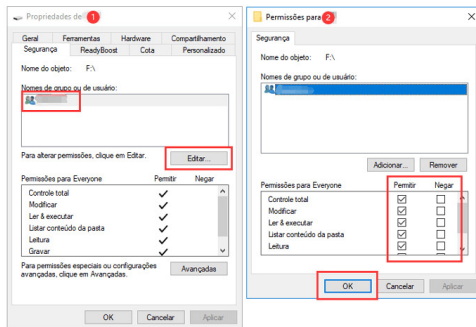
Escalada de privilégios



Escalada de privilégios (Windows).
Fonte: [PasswordRecovery, 2018]

Permissão de Acesso a Arquivos

- **Windows:** arquivos do sistema operacional e administradores são restritos, por padrão, para usuários comuns e convidados
 - Configurações adicionais podem ser incluídas nos arquivos, para proteção e permissão extra.



Escalada de privilégios (Windows).

Fonte: [EaseUS, 2021]

Permissão de Acesso a Arquivos

- **Linux:** cada arquivo contém um código de 9 bits binários para proteção de arquivos e/ou pastas;
 - Cada código contém 3 campos de 3 bits;
 - Os três primeiros bits são destinados ao proprietário do arquivo, os próximos três aos membros do grupo do proprietário e três últimos aos demais usuários;
 - Cada campo possui um bit para permissão de leitura (**r**), um bit para escrita (**w**) e outro para execução do programa (**x**);
 - Ex.: permissão ***rwxr-x--x***
 - Proprietário: pode ler, gravar e executar o arquivo;
 - Grupo: pode ler e executar o arquivo;
 - Demais usuários: podem somente executar o arquivo.

Muitos sistemas Linux formatam a proteção de arquivos utilizando números. A permissão 777 indica que todos os usuários podem ler, escrever e executar o arquivo.

Escalada de privilégios

```
jpp@jpp:/boot
jpp@jpp:/boot$ ls -la
total 39132
drwxr-xr-x  3 root root    4096 2011-05-13 08:52 .
drwxr-xr-x 23 root root    4096 2011-05-04 09:27 ..
-rw-r--r--  1 root root  700761 2011-03-18 16:33 abi-2.6.35-28-generic
-rw-r--r--  1 root root  730039 2011-04-11 01:24 abi-2.6.38-8-generic
-rw-r--r--  1 root root  122616 2011-03-18 16:33 config-2.6.35-28-generic
-rw-r--r--  1 root root  130313 2011-04-11 01:24 config-2.6.38-8-generic
drwxr-xr-x  3 root root    12288 2011-05-04 09:32 grub
-rw-r--r--  1 root root 11008098 2011-04-15 08:58 initrd.img-2.6.35-28-generic
-rw-r--r--  1 root root 13134896 2011-05-13 08:52 initrd.img-2.6.38-8-generic
-rw-r--r--  1 root root  160988 2010-10-22 09:08 memtest86+.bin
-rw-r--r--  1 root root  163168 2010-10-22 09:08 memtest86+ multiboot.bin
-rw-r--r--  1 root root  2344143 2011-03-18 16:33 System.map-2.6.35-28-generic
-rw-r--r--  1 root root  2654256 2011-04-11 01:24 System.map-2.6.38-8-generic
-rw-r--r--  1 root root    1336 2011-03-18 16:35 vmcoreinfo-2.6.35-28-generic
-rw-r--r--  1 root root    1368 2011-04-11 01:26 vmcoreinfo-2.6.38-8-generic
-rw-r--r--  1 root root 4342384 2011-03-18 16:33 vmlinuz-2.6.35-28-generic
-rw-r--r--  1 root root 4523936 2011-04-11 01:24 vmlinuz-2.6.38-8-generic
jpp@jpp:/boot$
```



Permissão de Acesso a Arquivos (Linux).

Fonte: [Nguyen, 2020]

Muitos sistemas Linux formatam a proteção de arquivos utilizando números. A permissão 777 indica que todos os usuários podem ler, escrever e executar o arquivo.

CHAMADAS DE SISTEMA (SYSTEM CALLS)

Chamadas de Sistema (System Calls)

- Sistemas operacionais possuem como uma de suas funções a capacidade de prover abstrações para programas de usuários
 - Essas abstrações permitem que os programas acessem recursos computacionais com maior facilidade;
 - Não é necessário que os programas possuam códigos fontes para comunicação com diferentes hardwares;
 - O gerenciamento de recursos do SO é transparente para usuários e feito automaticamente;
 - Programas “conversam” apenas com uma interface do SO, que provê acesso aos recursos desejados;
 - A troca de informações entre programas e o SO é feita por meio de **chamadas de sistema** [Tanenbaum and Bos, 2014].

System Calls

- Chamadas do sistema variam entre sistemas operacionais
 - No entanto, os conceitos são similares e podem ser aproveitados para diferentes sistemas operacionais;
 - Devido ao padrão POSIX⁷, sistemas operacionais modernos, como UNIX, System V, BSD, MINIX e Linux, utilizam uma interface similar [Tanenbaum and Bos, 2014];
 - Com isso, é possível fazer chamadas de sistema a partir de comandos disponíveis em diversas linguagens de programação, como C/C++.

⁷ Acrônimo para *Portable Operating System Interface*. Segue o padrão internacional ISO/IEC 9945-1.

System Calls

- Quando um processo está executando um programa de usuário e precisa de um serviço do sistema, deve executar uma instrução para transferir o controle para o sistema operacional
- O sistema operacional analisa a chamada de sistema e executa-a, caso autorizado
 - Após a execução da chamada, o controle é retornado para o processo;
 - A execução, para o programa, pode ser interpretada simplesmente como uma chamada convencional, semelhante à chamada de um procedimento qualquer;
 - *Traps*⁸ alteram o modo de execução do processo do modo usuário para o modo kernel [Tanenbaum and Bos, 2014].

⁸Traps correspondem a exceções ou falhas, geradas pela CPU, que requerem mudança para o modo kernel, para gerenciamento da condição excepcional [Bower, 2015].

System Calls

- As tabelas abaixo contém exemplos de chamada de sistema, para gerenciamento de processos e de arquivos.

Process management

Call	Description
<code>pid = fork()</code>	Create a child process identical to the parent
<code>pid = waitpid(pid, &statloc, options)</code>	Wait for a child to terminate
<code>s = execve(name, argv, environp)</code>	Replace a process' core image
<code>exit(status)</code>	Terminate process execution and return status

File management

Call	Description
<code>fd = open(file, how, ...)</code>	Open a file for reading, writing, or both
<code>s = close(fd)</code>	Close an open file
<code>n = read(fd, buffer, nbytes)</code>	Read data from a file into a buffer
<code>n = write(fd, buffer, nbytes)</code>	Write data from a buffer into a file
<code>position = lseek(fd, offset, whence)</code>	Move the file pointer
<code>s = stat(name, &buf)</code>	Get a file's status information

Chamadas de Sistema padrão POSIX (Gerenciamento de Processos e de Arquivos).

Fonte: [Silberschatz et al., 2012]

System Calls

- As tabelas abaixo contém exemplos de chamada de sistema.

Directory- and file-system management

Call	Description
s = mkdir(name, mode)	Create a new directory
s = rmdir(name)	Remove an empty directory
s = link(name1, name2)	Create a new entry, name2, pointing to name1
s = unlink(name)	Remove a directory entry
s = mount(special, name, flag)	Mount a file system
s = umount(special)	Unmount a file system

Miscellaneous

Call	Description
s = chdir(dirname)	Change the working directory
s = chmod(name, mode)	Change a file's protection bits
s = kill(pid, signal)	Send a signal to a process
seconds = time(&seconds)	Get the elapsed time since Jan. 1, 1970

Chamadas de Sistema padrão POSIX (Gerenciamento de Diretórios e Funções Úteis).

Fonte: [Silberschatz et al., 2012]

System Calls

- A tabela abaixo compara chamadas de sistema no padrão POSIX e Win32.

UNIX	Win32	Description
fork	CreateProcess	Create a new process
waitpid	WaitForSingleObject	Can wait for a process to exit
execve	(none)	CreateProcess = fork + execve
exit	ExitProcess	Terminate execution
open	CreateFile	Create a file or open an existing file
close	CloseHandle	Close a file
read	ReadFile	Read data from a file
write	WriteFile	Write data to a file
lseek	SetFilePointer	Move the file pointer
stat	GetFileAttributesEx	Get various file attributes
mkdir	CreateDirectory	Create a new directory
rmdir	RemoveDirectory	Remove an empty directory
link	(none)	Win32 does not support links
unlink	DeleteFile	Destroy an existing file
mount	(none)	Win32 does not support mount
umount	(none)	Win32 does not support mount, so no umount
chdir	SetCurrentDirectory	Change the current working directory
chmod	(none)	Win32 does not support security (although NT does)
kill	(none)	Win32 does not support signals
time	GetLocalTime	Get the current time

Chamadas de Sistema padrão POSIX e Win32.

Fonte: [Silberschatz et al., 2012]

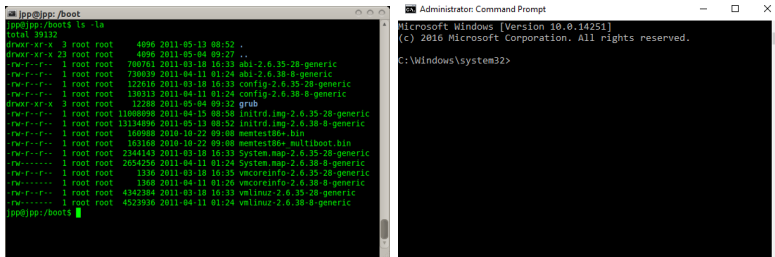
SHELL E GRAPHICAL USER INTERFACE (GUI)

Shell

- Sistemas operacionais são os responsáveis pelo gerenciamento e execução de chamadas de sistemas;
 - Essas chamadas podem ser realizadas pelos processos em execução;
 - As chamadas podem também ser invocadas por interpretadores de comandos - denominados **shell** (UNIX) ou **prompt de comandos** (Windows).
 - Salienta-se que tanto o *shell* quando o *command prompt* não são partes do sistema operacional
 - Podem ser consideradas apenas interfaces para uso de recursos do sistema operacional [Tanenbaum and Bos, 2014].

Shell

- O shell Linux e o prompt de comandos do Windows podem ser vistos na figura abaixo.



```

jpp@jpp:/boot
jpp@jpp:/boot$ ls -la
total 39132
drwxr-xr-x 3 root root 4096 2011-05-13 08:52 .
drwxr-xr-x 23 root root 4096 2011-05-04 09:27 ..
-rw-r--r-- 1 root root 780761 2011-03-18 16:33 abi-2.6.35-28-generic
-rw-r--r-- 1 root root 730039 2011-04-11 01:24 abi-2.6.38-8-generic
-rw-r--r-- 1 root root 122616 2011-03-18 16:33 config-2.6.35-28-generic
-rw-r--r-- 1 root root 130313 2011-04-11 01:24 config-2.6.38-8-generic
drwxr-xr-x 3 root root 12288 2011-05-04 09:32 grub
-rw-r--r-- 1 root root 11088098 2011-04-15 08:58 initrd.img-2.6.35-28-generic
-rw-r--r-- 1 root root 13134896 2011-05-13 08:52 initrd.img-2.6.38-8-generic
-rw-r--r-- 1 root root 160988 2010-10-22 09:08 memtest86+.bin
-rw-r--r-- 1 root root 163168 2010-10-22 09:08 memtest86+ multiboot.bin
-rw-r--r-- 1 root root 2344143 2011-03-18 16:33 System.map-2.6.35-28-generic
-rw-r--r-- 1 root root 2654256 2011-04-11 01:24 System.map-2.6.38-8-generic
-rw-r--r-- 1 root root 1336 2011-03-18 16:35 vmcoreinfo-2.6.35-28-generic
-rw-r--r-- 1 root root 1368 2011-04-11 01:26 vmcoreinfo-2.6.38-8-generic
-rw-r--r-- 1 root root 4342384 2011-03-18 16:33 vmlinuz-2.6.35-28-generic
-rw-r--r-- 1 root root 4523936 2011-04-11 01:24 vmlinuz-2.6.38-8-generic
jpp@jpp:/boot$

Administrator: Command Prompt
Microsoft Windows [Version 10.0.14251]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
  
```

Terminais no Linux e Windows.

Fonte: [Nguyen, 2020] e adaptado de [Winaero, 2015]

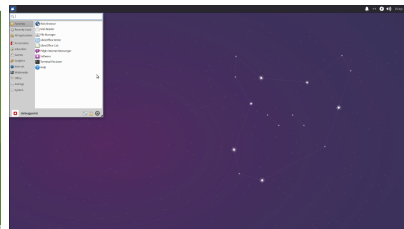
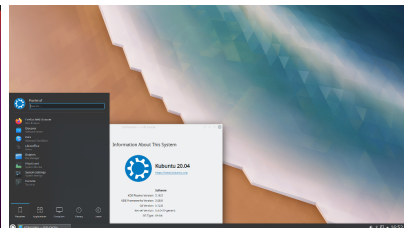
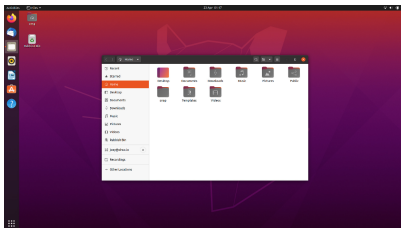
O shell Linux é frequentemente denominado “Terminal”.

Graphical User Interface (GUI)

- As chamadas de sistemas podem também ser realizadas por interfaces gráficas (GUI);
- Interfaces gráficas facilitam o uso de sistemas operacionais, tornando-os mais convenientes
 - Sistemas Linux permitem a instalação e seleção de diferentes interfaces gráficas, que podem ser escolhidas pelo usuário;
 - Sistemas Windows e Mac OS não possibilitam a instalação de novas interfaces⁹;
- Em alguns cenários, entretanto, podem ser consideradas desnecessárias, devido ao gasto de memória e de processamento computacional.

⁹Não confundir interface com skins, disponíveis para sistema Windows. Skins correspondem a atualizações apenas visuais na interface, não mudando o comportamento do sistema.

Graphical User Interface (GUI)



Interface gráficas GNOME 3, KDE, Mate e XFCE no Ubuntu.

Fonte: [Martins, 2020], [Wikipedia Contributors, 2021a], [Wikipedia Contributors, 2021b] e [DebugPoint, 2020]

Referências I



Bower, T. (2015).

Basics of how operating systems work.

[Online]; acessado em 06 de Setembro de 2021. Disponível em:

<http://faculty.salina.k-state.edu/tim/ossg/Introduction/OSworking.html>.



Computer Hope (2018).

Directory.

[Online]; acessado em 29 de Abril de 2021. Disponível em:

<https://www.computerhope.com/jargon/d/director.htm>.



DebugPoint (2020).

What's new in xubuntu 20.04 â feature tour.

[Online]; acessado em 17 de Maio de 2021. Disponível em:

<https://www.debugpoint.com/2020/04/xubuntu-2-04-release-download/>.



EaseUS (2021).

O windows não pode acessar o dispositivo caminho ou arquivo especificado.

[Online]; acessado em 16 de Maio de 2021. Disponível em:

<https://br.easeus.com/data-recovery-solution/windows-nao-pode-acessar-dispositivo-caminho-arquivo-especificado.html>.



Lucideus - Medium (2018).

Privilege escalation on windows 7, 8,10.

[Online]; acessado em 15 de Abril de 2020. Disponível em: <https://medium.com/@lucideus/privilege-escalation-on-windows-7-8-10-lucideus-research-c8a24aa55679>.

Referências II



Martins, C. (2020).

Atualizando do ubuntu 18.04? veja esses novos recursos.

[Online]; acessado em 17 de Maio de 2021. Disponível em:

<https://sempreupdate.com.br/atualizando-do-ubuntu-18-04-veja-esses-novos-recursos/>.



Nguyen, L. (2020).

Linux basics: File permissions.

[Online]; acessado em 16 de Maio de 2021. Disponível em:

<https://medium.com/@thisislong/linux-basics-file-permissions-792dbd7790ce>.



PasswordRecovery (2018).

How to open notepad as administrator in windows 10.

[Online]; acessado em 18 de Abril de 2020. Disponível em:

<https://www.top-password.com/blog/tag/run-notepad-as-administrator/>.



Silberschatz, A., Galvin, P. B., and Gagne, G. (2012).

Operating System Concepts.

Wiley Publishing, 9th edition.



Tanenbaum, A. S. and Bos, H. (2014).

Modern Operating Systems.

Prentice Hall Press, USA, 4th edition.



The Geek Page (2016).

How to end process in task manager of windows 10.

[Online]; acessado em 14 de Abril de 2020. Disponível em:

<https://thegeekpage.com/how-to-end-process-in-task-manager-of-windows-10/>.

Referências III



Wikipedia Contributors (2019).

Sistema de ficheiros.

[Online]; acessado em 14 de Abril de 2020. Disponível em:
https://pt.wikipedia.org/wiki/Sistema_de_ficheiros.



Wikipedia Contributors (2021a).

Kubuntu.

[Online]; acessado em 17 de Maio de 2021. Disponível em: <https://pt.wikipedia.org/wiki/Kubuntu>.



Wikipedia Contributors (2021b).

Ubuntu mate.

[Online]; acessado em 17 de Maio de 2021. Disponível em:
https://pt.wikipedia.org/wiki/Ubuntu_MATE.



Winaero (2015).

How to open elevated command prompt in windows 10.

[Online]; acessado em 17 de Maio de 2021. Disponível em:
<https://winaero.com/how-to-open-elevated-command-prompt-in-windows-10>.