Curso 4451

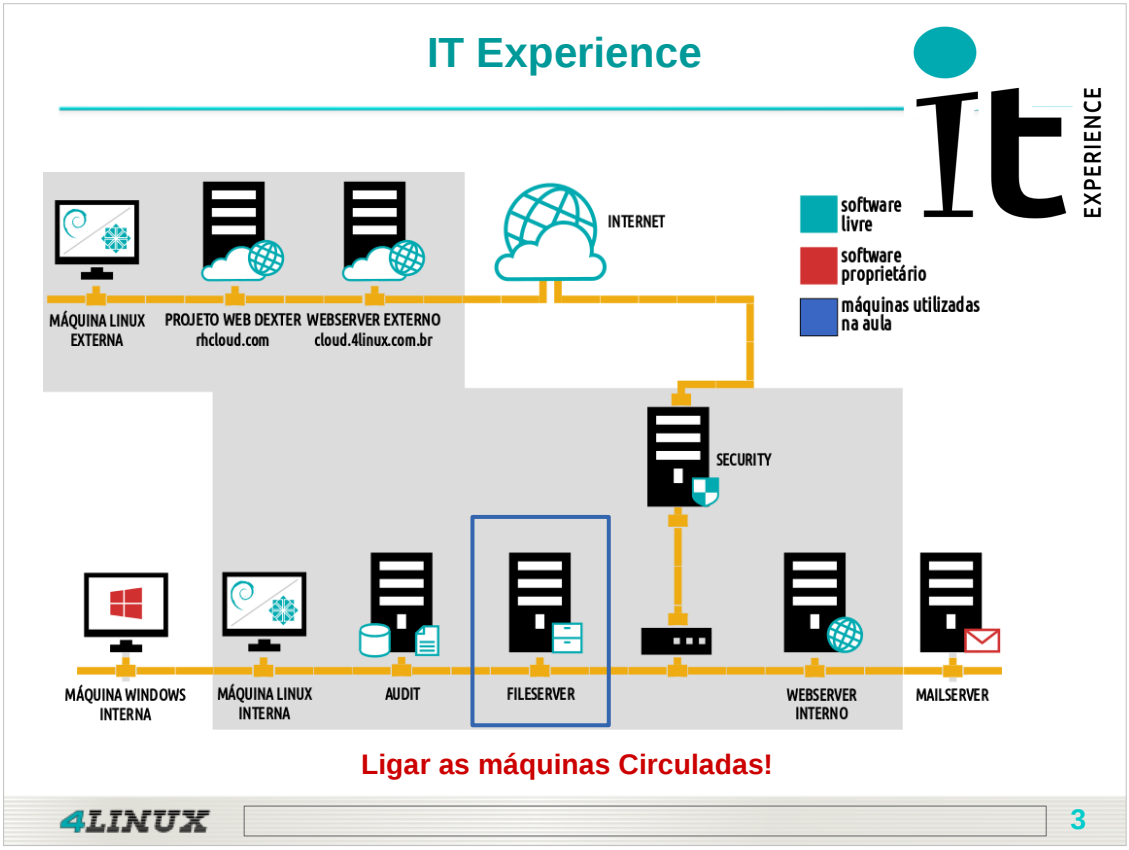
Linux Security Administration in Cloud

Registro de usuários no sistema

Quando começamos a trabalhar com usuários no sistema GNU/Linux podemos dividi-los em três categorias:

• **Usuário Administrador** (Super Usuário): usuário conhecido como “root” no sistema. É esse usuário que controla todo o sistema e não possui nenhuma restrição. Mas devemos ter uma certa cautela ao usá-lo pois com qualquer deslize podemos danificar todo o sistema;

• **Usuários de Sistema:** são aqueles que não precisam “logar” no sistema, são utilizados para controlar serviços. Esses usuários não devem possuir senhas nem “Shell” válida. Um exemplo desses usuários é o “www-data” que é usado exclusivamente para controlar o servidor web “Apache”;


IT Experience
EXPERIENCE
INTERNET
software livre
software proprietário
máquinas utilizadas na aula
MÁQUINA LINUX EXTERNA
PROJETO WEB DEXTER rhcloud.com
WEBSERVER EXTERNO cloud.4linux.com.br
SECURITY
MÁQUINA WINDOWS INTERNA
MÁQUINA LINUX INTERNA
AUDIT
FILESERVER
WEBSERVER INTERNO
MAILSERVER
Ligar as máquinas Circuladas!
4LINUX
3

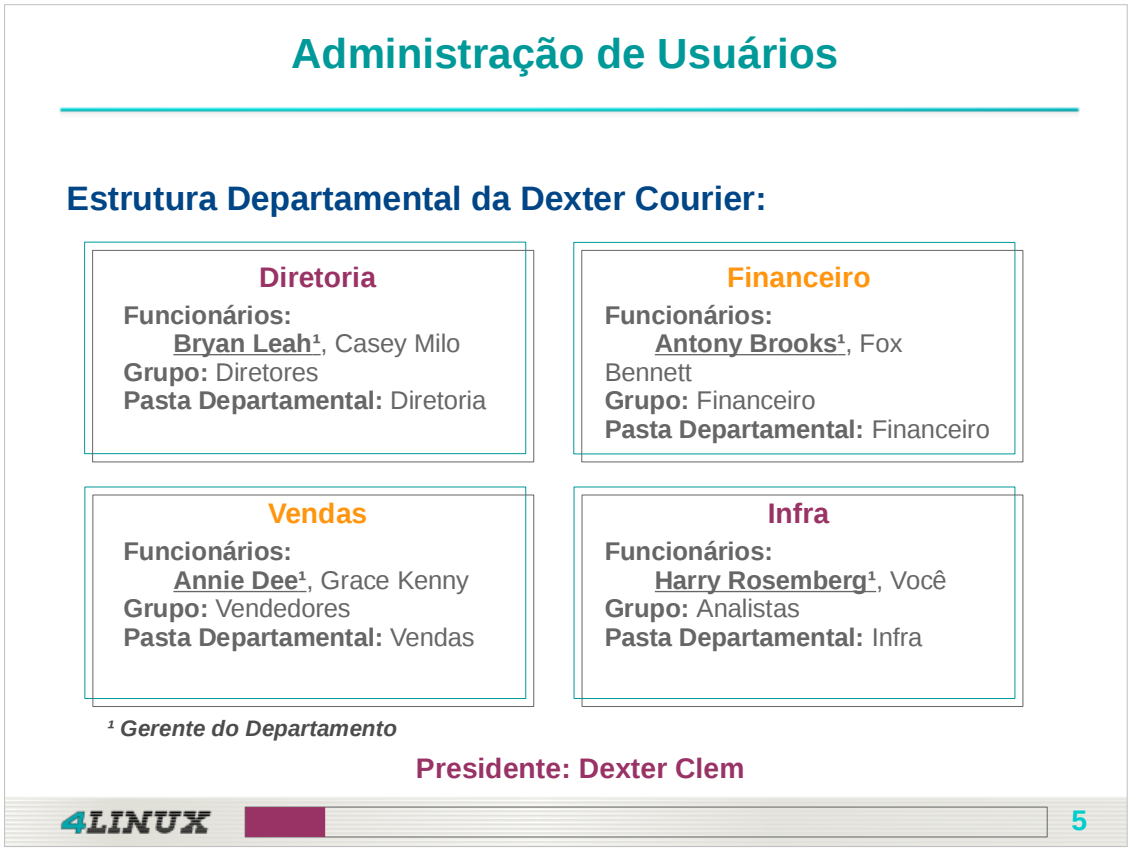
Anotações:

Objetivos da Aula

Aula 01 (1/2)

- Explorar os arquivos responsáveis pela administração de usuários e grupos;
- Aprender a administrar usuários, grupos e senhas de acesso;
- Automatizar através de scripts o processos de criação de usuários;
- Criar pastas departamentais e permissões específicas de acesso de acordo com o cenário da empresa Dexter.

Anotações:



• **Usuários comuns:** são utilizados para trabalhar no sistema GNU/Linux. São contas criadas para aqueles que utilizam ou operam o sistema. É sempre aconselhável que cada usuário comum ou administrador tenha sua própria conta e só utilize a conta “root” para administração do sistema.

Administração de Usuários

Regras Referente ao Gerenciamento de Usuários:

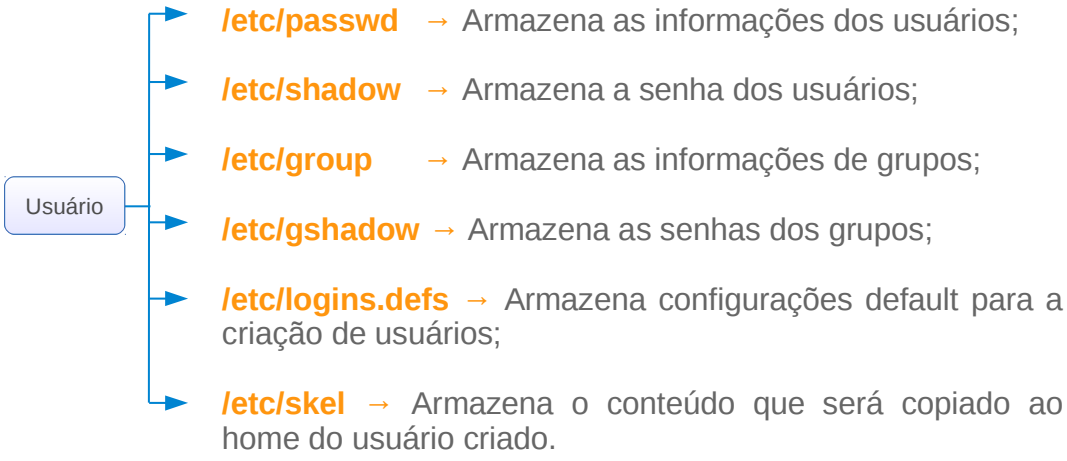
- **Política de Senha:**
  Expirar a cada 30 dias, mínimo 6 caracteres;
- **Pastas Pessoais:**
  Armazenar em /srv/homes/
  Conter os Diretórios: Documentos, Downloads, Imagens
- **Pastas Departamentais:**
  Armazenar em /srv/dexter/
- **Padrão de Nome de Usuários:**
  Login do Usuário: nome.sobrenome
  Senha Padrão: dexter
  Presidente: Login apenas com o nome sem sobrenome

Anotações:

Administração de Usuários

Servidor: FileServer

Arquitetura de Arquivos na Criação de um Usuário:



Anotações:

Administração de Usuários

Servidor: FileServer

Coletando Informações de Usuários Existentes:

1 – Verifique os arquivos passwd e group:

```
1# getent passwd
2# getent group
```

2 – Faça um levantamento de informações sobre o usuário suporte:

```
3# id suporte
4# groups suporte
```

O comando **getent** tem a função de listar todos os usuários existentes no sistema...

O comando **id** servirá simplesmente para verificar o id de um usuário e seus respectivos grupos.

Já o comando **groups** verifica quais são os grupos aos quais um usuário está vinculado.

Anotações:

Administração de Usuários

Servidor: FileServer

Coletando Informações de Usuários Existentes:

3 – Utilize o programa finger para detalhar informações sobre o usuário:

```
1# finger suporte
```

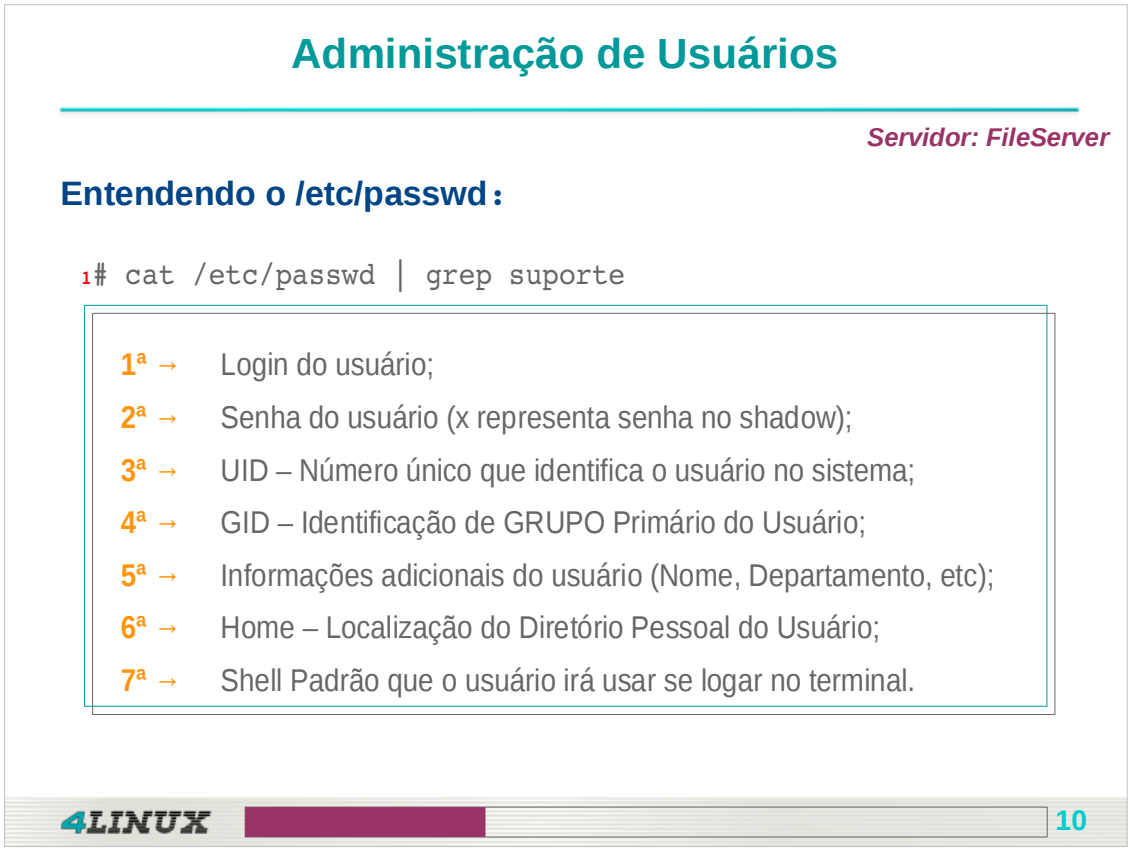
4 – Utilize os comandos who e w para verificar os usuários conectados:

```
2# who
3# w
```

O comando **finger** não vem instalado por padrão na maioria das distribuições, sua função é mostrar de maneira mais formatada as informações dos usuários.

O comando **who** permite verificar o nome do usuário logado enquanto o comando **"w"** verifica todos os usuários logados na máquina.

Anotações:



PWCONV e PWUNCONV

Caso encontremos algum servidor GNU/Linux sem as senhas “shadow” configuradas, podemos utilizar o comando **“pwconv”** para ativá-las e **“pwunconv”** para desativá-las.

O comando **“pwconv”** é usado para criar o arquivo shadow a partir do arquivo /etc/passwd , enquanto o comando **“pwunconv”** executa a operação inversa.

Administração de Usuários

UID 0: É o número do usuário administrador "root"

UID Debian:

1 a 999: Usuários de sistema

1000 a 65535: Usuários normais

UID CentOS:

1 a 999: Usuários de sistema

1000 a 65535: Usuários normais

GID 0: É o número do grupo administrador "root".

GID Debian:

1 a 999: Grupos de sistema

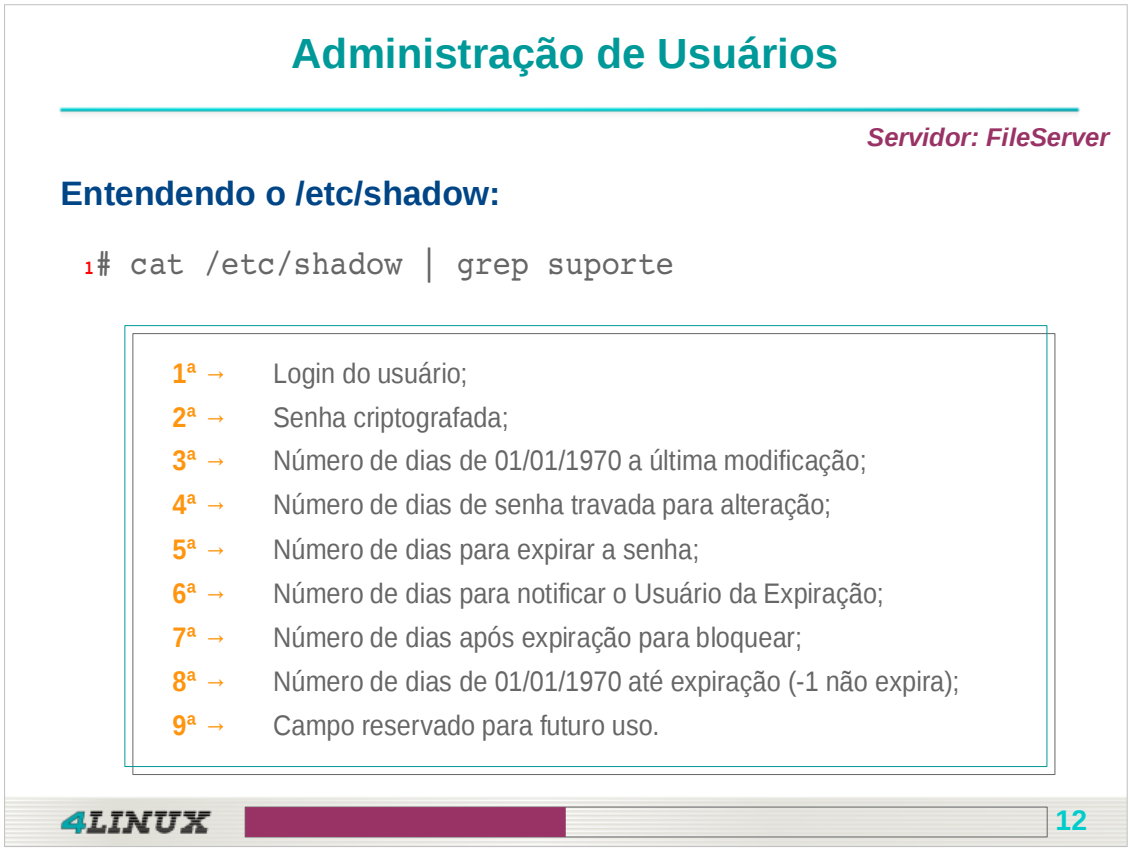
1000 a 65535: Grupos normais

GID CentOS:

1 a 999: Grupos de sistema

1000 a 65535: Grupos normais

Anotações:



Arquivo /etc/shadow

As senhas dos usuários ficam armazenadas no arquivo “/etc/shadow” conhecido como “senhas sombras” (shadow passwords). As senhas ficam nele pois é um arquivo mais seguro que o arquivo “/etc/passwd”. No arquivo “/etc/passwd” qualquer usuário poderia visualizá-las e copiá-las para outro diretório ou máquina remota.

Já o arquivo “/etc/shadow” tem suas permissões muito mais restritas, não permitindo que ele seja copiado e nem visualizado diretamente por um usuário comum. Isso é uma grande ajuda na questão de segurança, pois se as senhas estivessem no próprio ”/etc/passwd” seria muito fácil para um invasor com usuário comum, copiar esse arquivo para
outro servidor e aplicar uma ferramenta de “brute force” para quebrar as senhas.

Administração de Usuários

Servidor: FileServer

Adicionando os Usuários da Dexter:

```
1# adduser dexter
2# adduser bryan.leah
3# passwd bryan.leah (Senha: dexter)
```

Não adicione os demais ainda!!
Vamos checar algumas informações

```
4# tail -n 2 /etc/passwd
5# tail -n 2 /etc/group
6# tail -n 2 /etc/shadow
7# ls -l /home
```

O comando **adduser** é usado para criar usuários tanto no CentOS quando no Debian. No **Debian** por padrão não aceita login com o "." (ponto), portanto é preciso usar a opção **--force-badname**.

Caso tenha criado o usuário de forma errada use:

 # userdel -r bryan.leah

Anotações:

Importante

Servidor: FileServer

Na regras gerais o home dos usuários precisam estar armazenados no /srv/homes e por padrão o comando adduser cria o home do usuário no /home.

Vamos corrigir isso!

```
1# ls /home
2# ls /srv/homes
```

vamos corrigir a /home de usuário com o usermod:

```
3# usermod -m -d /srv/homes/suporte suporte
4# ls /srv/homes
5# grep suporte /etc/passwd
6# usermod -c 'Analistas - Infra' suporte
```

O comando **usermod** altera informações de usuários sem precisar editar arquivos de configuração.
Prefira sempre usá-lo ao invés de editar diretamente o /etc/passwd, e se ainda assim tiver de editar o arquivo utilize o comando **vipw** para isso.

Anotações:

Administração de Usuários

Servidor: FileServer

1- Faça a correção para os usuários Dexter e Bryan antes de continuar:

```
1# usermod -m -d /srv/homes/dexter dexter
2# usermod -m -d /srv/homes/bryan.leah bryan.leah
3# ls /srv/homes
4# tail -n 3 /etc/passwd
```

O diretório home está sem as pastas da regra geral (Documentos, Downloads, Imagens):

```
5# ls /srv/homes/bryan.leah
6# ls /etc/skel
7# mkdir /etc/skel/{Documentos,Downloads,Imagens}
8# ls /etc/skel/
```


Anotações:

Administração de Usuários

Servidor: FileServer

- Vamos apagar o usuário Bryan e criá-lo novamente de forma correta já que o /etc/skel está configurado:

```
1# userdel -r  bryan.leah
2# ls -l /srv/homes/
3# adduser --home /srv/homes/bryan.leah  bryan.leah
4# ls /srv/homes/bryan.leah
5# passwd bryan.leah (Senha: dexter)
```


Anotações:

Administração de Usuários

Servidor: FileServer

```
1# ls -l /srv/homes/bryan.leah
2# su – bryan.leah
3# pwd
4# ls -l
5# exit
```

Usuário Criado com Sucesso seguindo as Normas da Dexter!!!

O próximo passo é definir uma maneira de alterar a home de todos os usuários já criados, para isso faremos uma automação do processo utilizado através de um script:

```
6# cat /root/funcionarios.txt
7# vim /root/funcionarios.txt  <Substitua para Seu Nome e Sobrenome>
8# cd ~
```


Anotações:

Administração de Usuários

Servidor: FileServer

Explorando Comandos para Criar um Shell Script:

1 – Testando substituições com o comando tr:

```
1# echo dexter | tr "e" "&"
2# cat funcionarios.txt
3# cat funcionarios.txt | tr ";" "\t"
```

2 – O “tr” também substitui o texto caixa alta (upper), por caixa baixa (lower) e vice-versa.

```
4# tr [:upper:] [:lower:] < funcionarios.txt
```

O comando **tr** é um comando usado para substituir (trocar) ocorrências.
Quando concatenado com o “|”, (pipe) as alterações são mostradas na saída padrão ao invés de serem gravadas.
De forma similar ao comando cpio, o tr sempre deverá receber uma determinada saída e atuar sobre ela, por isso o uso do redirecionar “<” (volta).

Anotações:

Administração de Usuários

Servidor: FileServer

Explorando Comandos para criar um Shell Script:

3 – Vamos testar o uso do cut para “cortar” informações de um arquivo:

```
1# cut -d";" -f1 funcionarios.txt
```

< Junte isso ao testes efetuados com o tr: >

```
2# cut -d";" -f1 funcionarios.txt | tr [:upper:] [:lower:]
3# cut -d";" -f1 funcionarios.txt | tr [:upper:] [:lower:] | tr " " "."
```

O comando **cut** permite extrair informações de uma determinada saída. Neste exemplo utilizamos como base um delimitador **“;”** definido pelo parâmetro **"-d"** e trouxemos o primeiro campo separado pelo delimitador **-"f1"**.

Anotações:

Administração de Usuários

Servidor: FileServer

1- Automatizando a criação dos demais usuários com Shell Script:

```
1# vim createuser.sh
1 #!/bin/bash
2 test -e /root/funcionarios.txt || exit
3 while read FUNCIONARIO
4 do
5  NOME=$(echo  $FUNCIONARIO  |  cut  -d";"  -f1  |  tr  [:upper:]
[:lower:] | tr " " ".")
6 DEPARTAMENTO=$(echo $FUNCIONARIO | cut -d";" -f2)
7 useradd -m -b /srv/homes -c "$FUNCIONARIO" -s /bin/bash $NOME
8 echo -e "dexter\ndexter" | passwd $NOME
9 done < /root/funcionarios.txt

2# bash /root/createuser.sh
```


Anotações:

Administração de Usuários

Servidor: FileServer

Validando os Usuários Criados:

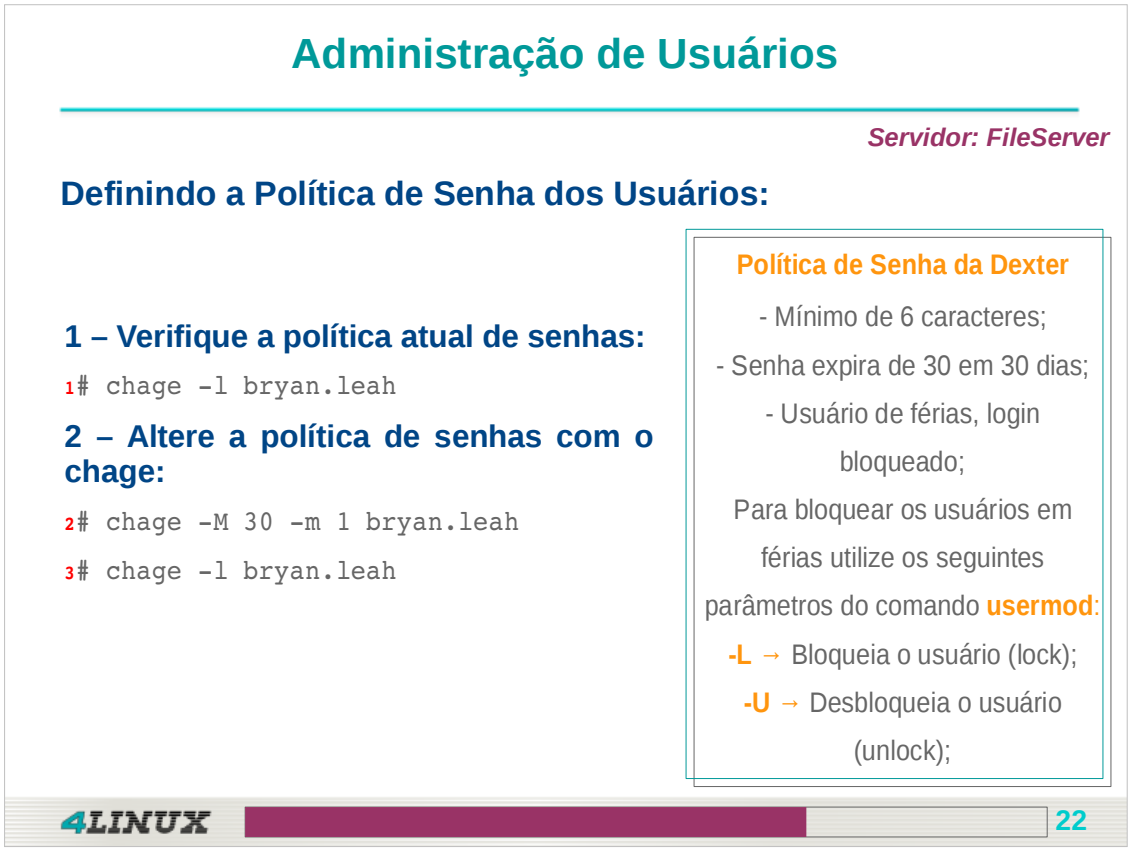
1 – Utilize o alias criado no 4450 para verificar os usuários criados:

```
1# users
```

2 – Verifique se as pastas de usuários foram criadas no /srv/homes:

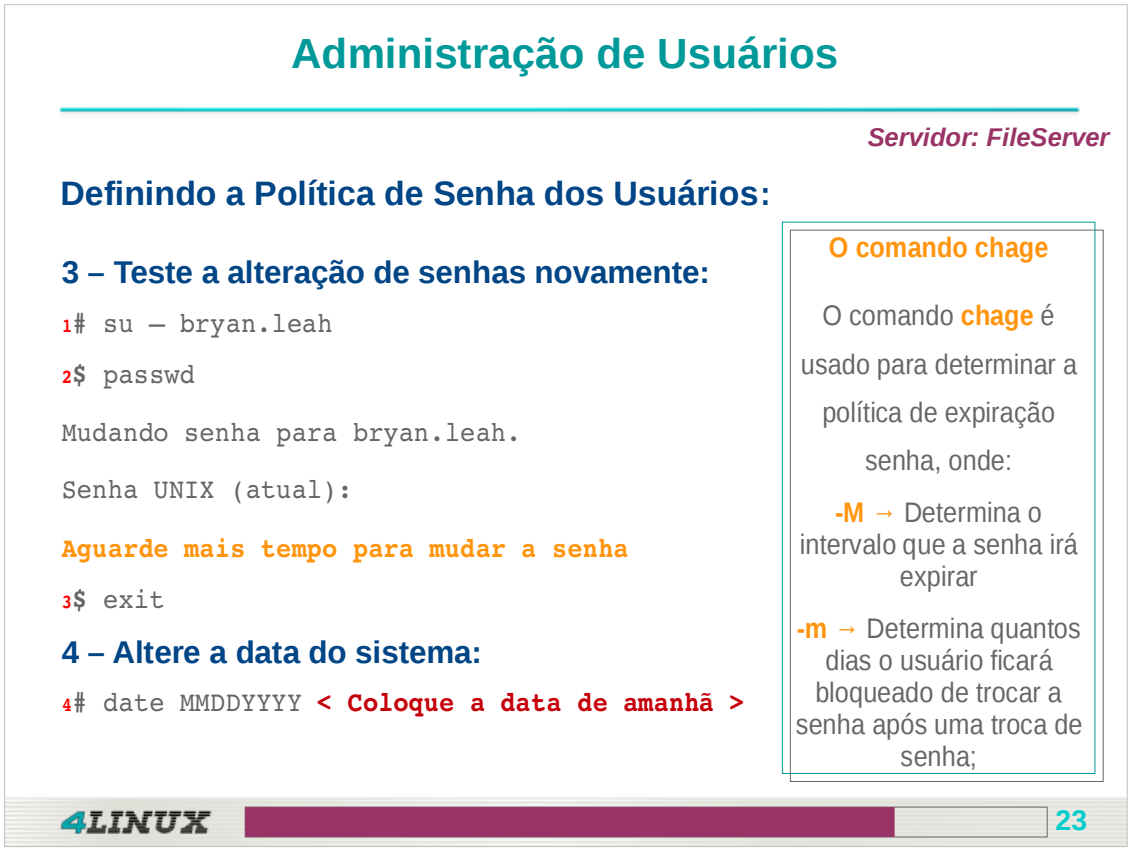
```
2# ls /srv/homes
3$ su – bryan.leah
4$ pwd
5# exit
6# finger bryan.leah
```


Anotações:



Comando Chage

O comando “chage” configura algumas características da senha, como: data de validade, data de aviso de troca, dentre outras opções do arquivo shadow, fique atento, porque esse comando é muito útil em seu dia-a-dia.



Alterações de senha com passwd

Caso esteja modificando a senha de um usuário normal, primeiro será solicitada a senha corrente para permitir a definição de uma nova senha. Isso não acontece com o usuário “root”, que pode definir a nova senha diretamente, tanto para ele quanto para os outros usuários.

Administração de Usuários

Servidor: FileServer

Definindo a Política de Senha dos Usuários:

4 – Logue com o usuário e tente novamente:

```
1# su – bryan.leah
2$ passwd
< Ele irá permitir a troca de senha >
< Tente utilizar a senha 123 >
3$ exit
```

5 – Simule a senha Expirada verificando a data e alterando para +1 dia:

```
4# chage -l bryan.leah
5# date MMDDYY
< Tente logar novamente em um terminal >
```

O comando passwd

O comando **passwd** por ser usado para definição de senhas ou para bloqueio de usuários:

-l → Bloqueia o usuário (lock)

-u → Desbloqueia o usuário (unlock);

Não esqueça de voltar a data ao normal depois dos testes!

Anotações:

Administração de Usuários

Servidor: FileServer

Adicionando os Grupos Departamentais:

```
1# groupadd diretores
2# groupadd vendedores
3# groupadd financeiro
4# groupadd analistas
5# tail -n 4 /etc/group
```

O comando **addgroup** é usado para criar grupos tanto no CentOS quando no Debian.

Caso tenha criado o grupo de forma errada use:

 # groupdel diretores

Anotações:

Administração de Usuários

Servidor: FileServer

Adicionando os Usuários nos Grupos Departamentais :

```
1# gpasswd -a bryan.leah diretores
2# gpasswd -a casey.milo diretores
3# gpasswd -a annie.dee vendedores
4# gpasswd -a grace.kenny vendedores
5# gpasswd -a antony.brooks financeiro
6# gpasswd -a fox.bennett financeiro
7# gpasswd -a harry.rosemberg analistas
8# gpasswd -a voce.sobrenome analistas
9# tail -n 4 /etc/group
```

O comando **gpasswd** é usado para adicionar ou remover usuários de grupos.

Caso tenha colocado um funcionário em um grupo errado use:

 # gpasswd -d user group

Anotações:

Administração de Usuários

Servidor: FileServer

Adicionando as Pastas Departamentais e Ajustando os Proprietários:

```
1# cd /srv/dexter
2# mkdir {Diretoria,Vendas,Financeiro,Infra}
3# ls -l
drwxr-xr-x 2 root root ... Diretoria
drwxr-xr-x 2 root root ... Financeiro
drwxr-xr-x 2 root root ... Infra
drwxr-xr-x 2 root root ... Vendas
4# chown bryan.leah Diretoria
5# chown annie.dee Vendas
6# chown antony.brooks Financeiro
7# chown harry.rosemberg Infra
```

Cada pasta departamental da Dexter terá seu gerente como dono e o departamento como grupo dono.

O comando **chown** é usado alterar o dono de pastas e arquivos, já o comando **chgrp** é usando para alterar o grupo dono de pastas e arquivos.

Anotações:

Administração de Usuários

Servidor: FileServer

Ajustando os Grupos das Pastas:

```
1# ls -l
drwxr-xr-x 2 bryan.leah      root ... Diretoria
drwxr-xr-x 2 antony.brooks root ... Financeiro
drwxr-xr-x 2 harry.rosemberg root ... Infra
drwxr-xr-x 2 annie.dee       root ... Vendas
2# chgrp diretores Diretoria
3# chgrp vendedores Vendas
4# chgrp financeiro Financeiro
5# chgrp analistas Infra
6# ls -l
drwxr-xr-x 2 bryan.leah          diretores  ...  Diretoria
drwxr-xr-x 2 antony.brooks    financeiro    ...  Financeiro
drwxr-xr-x 2 harry.rosemberg analistas  ...  Infra
drwxr-xr-x 2 annie.dee       vendedores    ...  Vendas
```

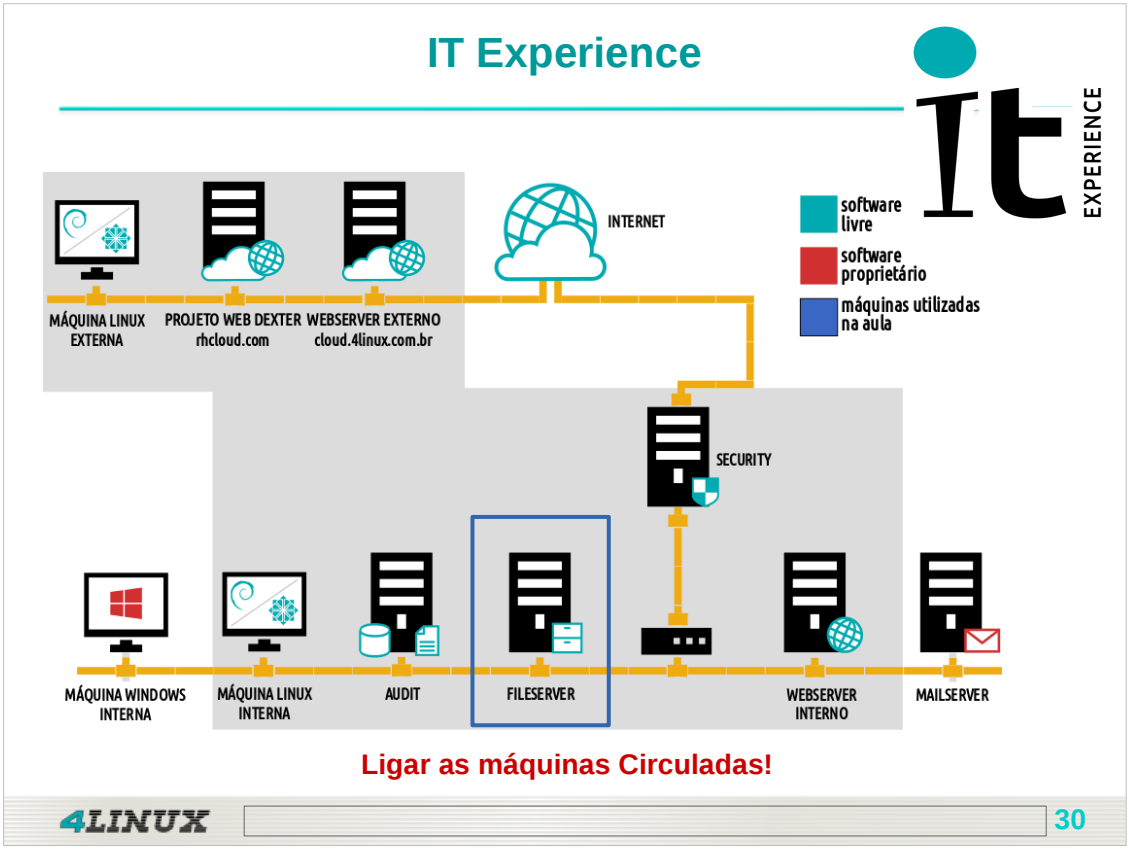

Anotações:

Registro de usuários no sistema

Quando começamos a trabalhar com usuários no sistema GNU/Linux podemos dividi-los em três categorias:

• **Usuário Administrador** (Super Usuário): usuário conhecido como “root” no sistema. É esse usuário que controla todo o sistema e não possui nenhuma restrição. Mas devemos ter uma certa cautela ao usá-lo pois com qualquer deslize podemos danificar todo o sistema;

• **Usuários de Sistema:** são aqueles que não precisam “logar” no sistema, são utilizados para controlar serviços. Esses usuários não devem possuir senhas nem “Shell” válida. Um exemplo desses usuários é o “www-data” que é usado exclusivamente para controlar o servidor web “Apache”;


IT Experience
IT EXPERIENCE
INTERNET
software livre
software proprietário
máquinas utilizadas na aula
MÁQUINA LINUX EXTERNA
PROJETO WEB DEXTER rhcloud.com
WEBSERVER EXTERNO cloud.4linux.com.br
SECURITY
MÁQUINA WINDOWS INTERNA
MÁQUINA LINUX INTERNA
AUDIT
FILESERVER
WEBSERVER INTERNO
MAILSERVER
Ligar as máquinas Circuladas!
4LINUX
30

Anotações:

Objetivos da Aula

Aula 01 (2/2)

- Alterar dono e grupo de arquivos e diretórios;
- Gerenciar permissões no sistema:
  - Literal e Octal;
  - Gerenciar permissões especiais.

Anotações:

Fundamentação

O GNU/Linux é um sistema multi-usuário e portanto, possui um esquema de permissões que provê a privacidade e/ou compartilhamento de arquivos entre usuários. Na verdade, esse esquema de permissões é parte fundamental do sistema.

Administração de Arquivos

Regras Referentes ao Gerenciamento de Arquivos:

- Todo departamento tem acesso de Leitura e Escrita em sua pasta departamental;
- Um departamento não pode acessar a pasta de outro departamento (**nem pra Leitura!!!**);
- Todos os arquivos criados na pasta do Departamento precisam ser do Grupo do departamento;
- Um usuário não pode apagar arquivos dentro da pasta do departamento, a não ser que ele seja o **DONO** do arquivo ou da pasta Departamental;
- O Presidente Dexter Clem acessa todas as pastas dos Departamentos.

Anotações:

Regras de Acesso a Arquivos

Servidor: FileServer

Analisando o Status Atual:

```
1# su – fox.bennett
2$ cd /srv/dexter/Financeiro
3$ touch relatorio.txt
touch: não foi possível tocar "relatorio.txt": Permissão negada
4$ cd /srv/dexter/Vendas
5$ pwd
6$ exit
```

Financeiro

Funcionários:

Antony Brooks[1], Fox Bennett

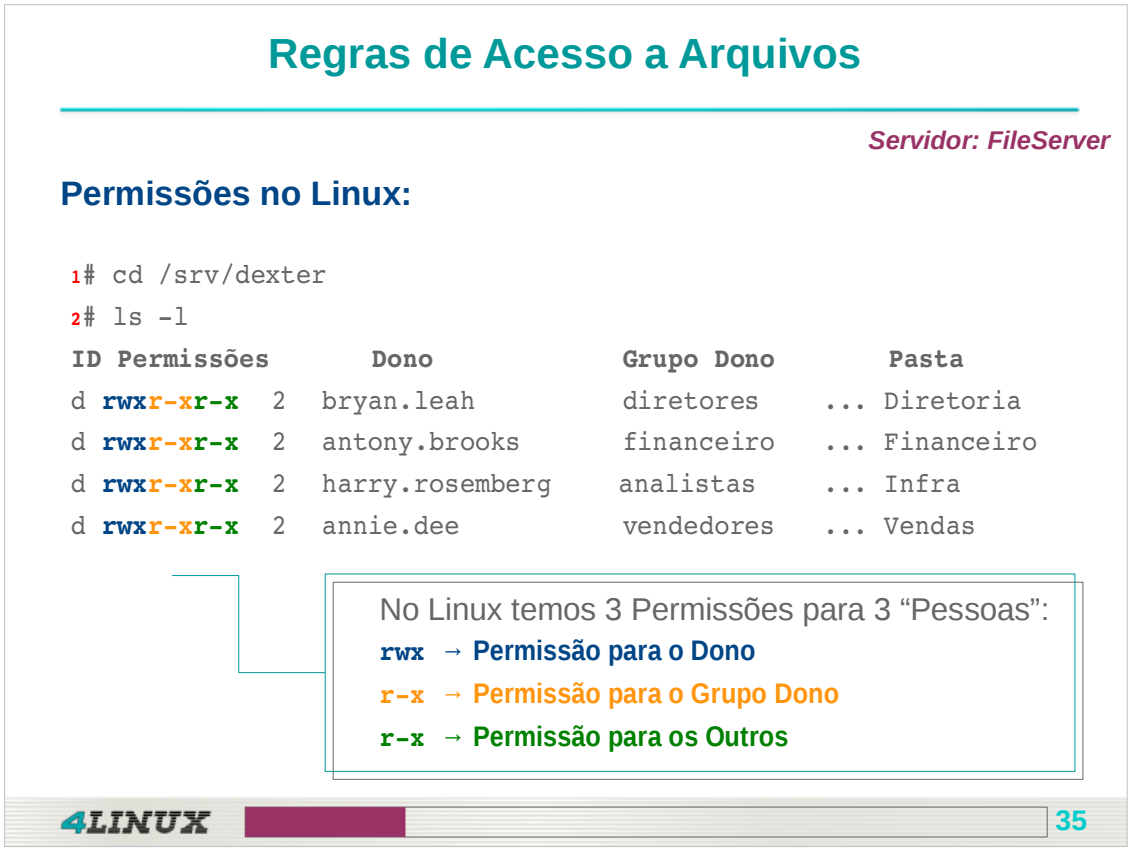
Grupo: Financeiro

Pasta Departamental: Financeiro

1° Problema:
O funcionário do departamento está sem acesso a criar arquivos.

2° Problema:
O funcionário consegue entrar em pastas de departamentos que ele não pertence.

Anotações:



Entendendo as permissões:

Ao utilizar o comando ls com o parâmetro menos "-l" uma listagem longa é exibida, nesta listagem o primeiro caractere pode ser:

"-" →indicando a listagem de um arquivo comum";
d → indicando um diretório;
l → indicando um "link" simbólico;
p → indicando um "pipe" nomeado;
s → indicando um "socket";
c → indicando um dispositivo de caractere;
b → indicando um dispositivo de bloco.

Regras de Acesso a Arquivos

Servidor: FileServer

Permissões no Linux:

"Pessoas"	Operadores	Permissões	Octal	Literal
u (user)	+	Leitura	4	r
g (group)	-	Escrita	2	w
o (others)	=	Execução[1]	1	x

[1] Permissão de Execução em Diretório significa entrar no diretório (cd)

Pastas Departamentais:

Dono = Leitura, Escrita e Execução → rwx → 7

Grupo = Leitura, Escrita e Execução → rwx → 7

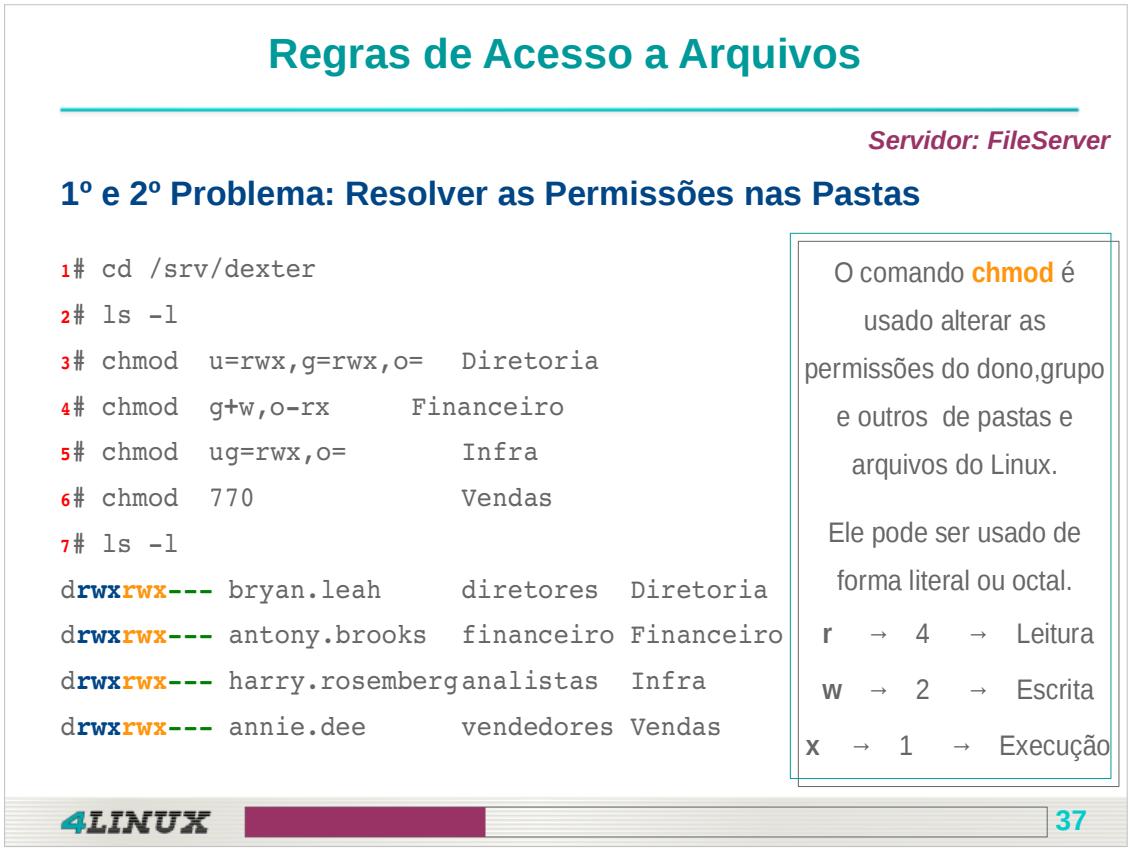
Outros = Nenhuma Permissão → --- → 0

4LINUX 36

Entendendo as permissões:

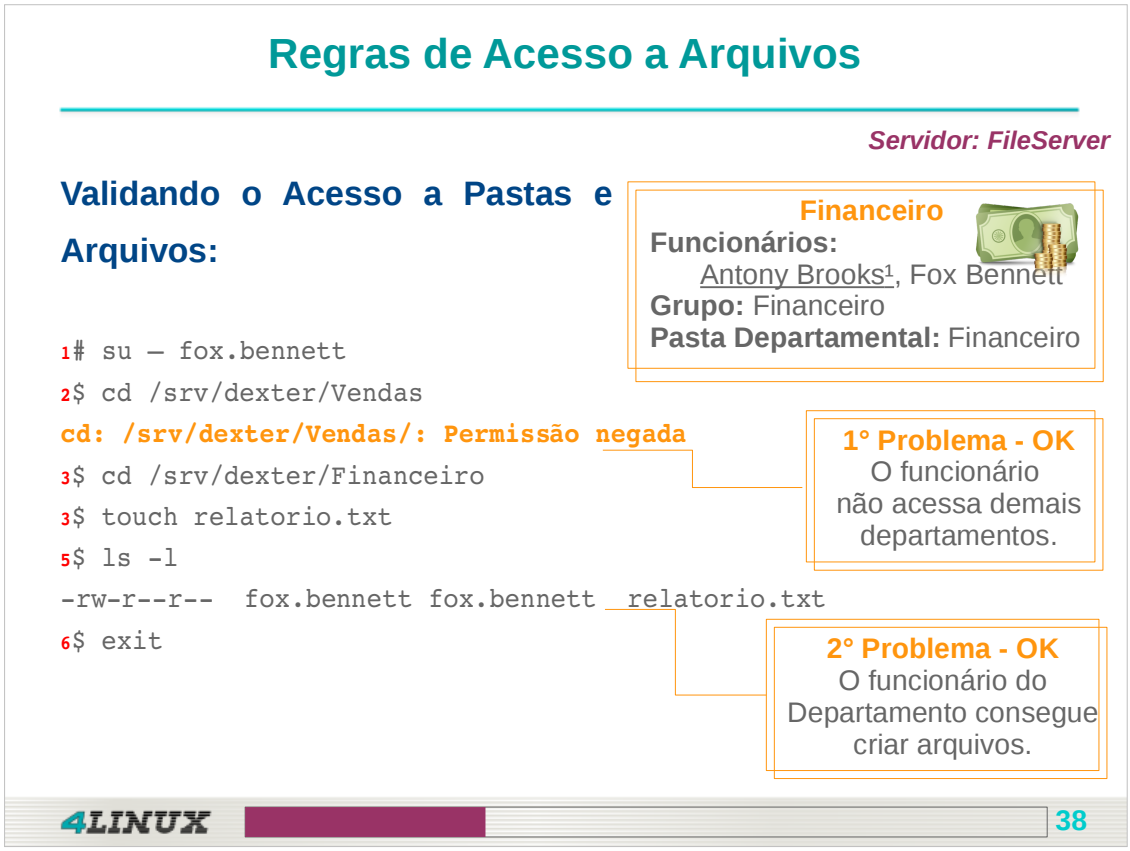
Após o primeiro carácter os próximos três conjuntos de três caracteres indicam as permissões do usuário dono do arquivo, permissões de grupo e permissões para outros usuários.

Nesses três conjuntos, se o caractere encontrado for um "-" (hífen) significa que a permissão está ausente, ou seja, não há a respectiva permissão. Se alguma ou todas as letras (r, w e x) forem encontradas, indicará as permissões que o arquivo tem permissões definidas conforme a tabela do slide acima.



Entendendo as permissões:

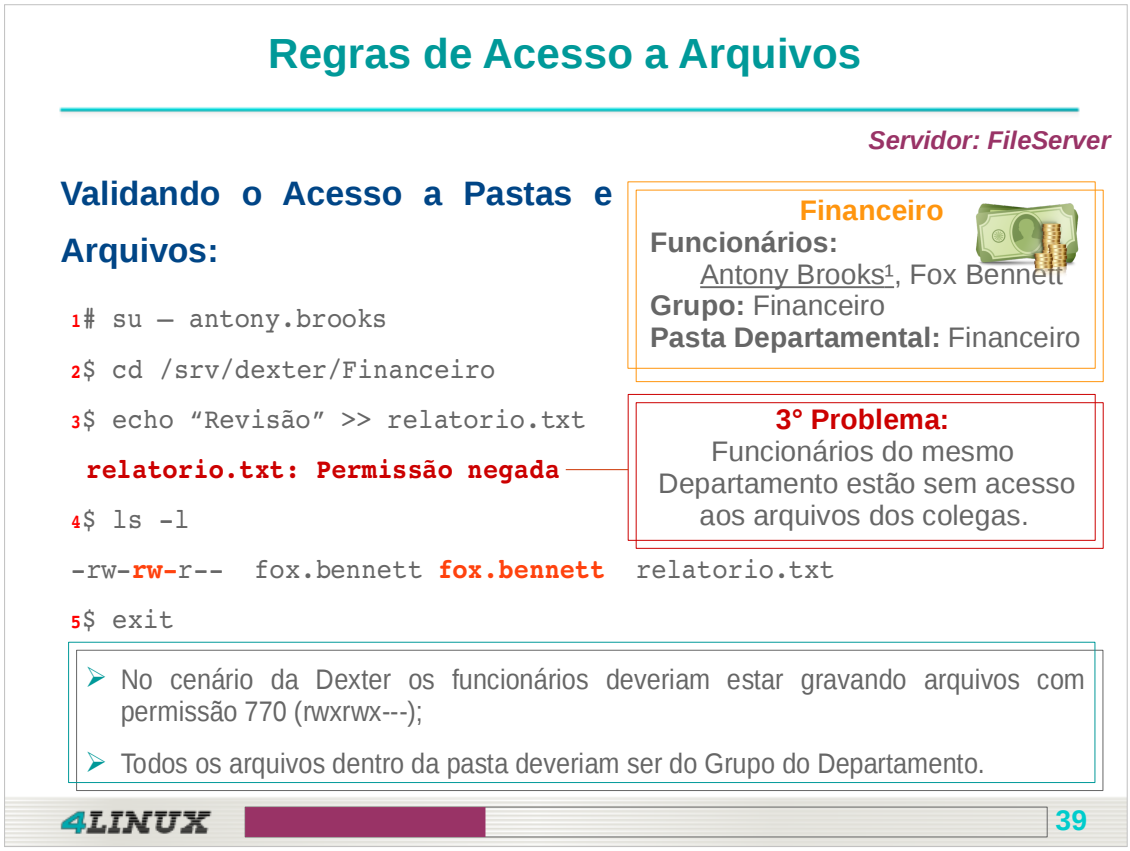
Seguindo o conjunto de permissões, há um número que indica a quantidade de “links” simbólicos que o arquivo ou diretório tem. Após o número de “links”, vem a indicação do usuário dono do arquivo, seguido do grupo ao qual ele, o arquivo ou diretório, pertence.



Atribuindo Permissões:

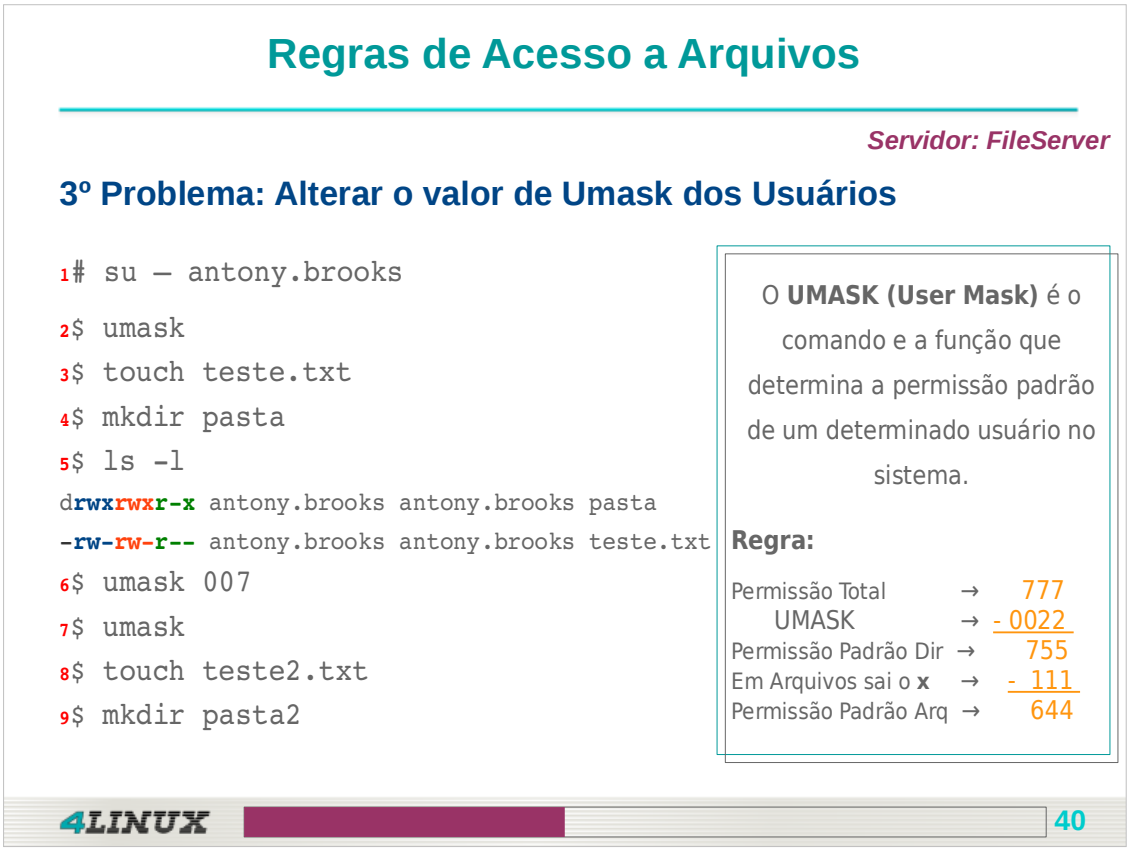
Os parâmetros “**u-rwx,g=rwx,o=**” usados no slide anterior definiram o esquema de permissões. A primeira letra indica para qual(is) usuário(s) as permissões estão sendo alteradas. Usamos a letra “u” para indicar o próprio dono, “g” para indicar o grupo, “o” para outros e ainda a letra “a” para indicar todos.

O caractere seguinte poderia ser um sinal de “=” para definir uma permissão “+” para adicionar valores a permissão ou “-” para retirar valores da permissão.



Resumo das Permissões no padrão Octal:

A sintaxe utilizada no momento em que definimos a permissão para o diretório Vendas utiliza o formato octal. Neste caso, o parâmetro que define as permissões é composto de três números de 0 a 7, que correspondem às permissões para o usuário dono, para o grupo e para outros. Cada número é formado pela soma das permissões atribuídas, sendo que execução vale 1, escrita vale 2 e leitura 4.



Permissionamento Umask

O “umask” altera o valor da máscara de criação de arquivos e diretórios. Essa “máscara” é utilizada para definir o “permissionamento” padrão de um arquivo ou diretório quando ele é criado.

Cálculo da umask d Diretório:

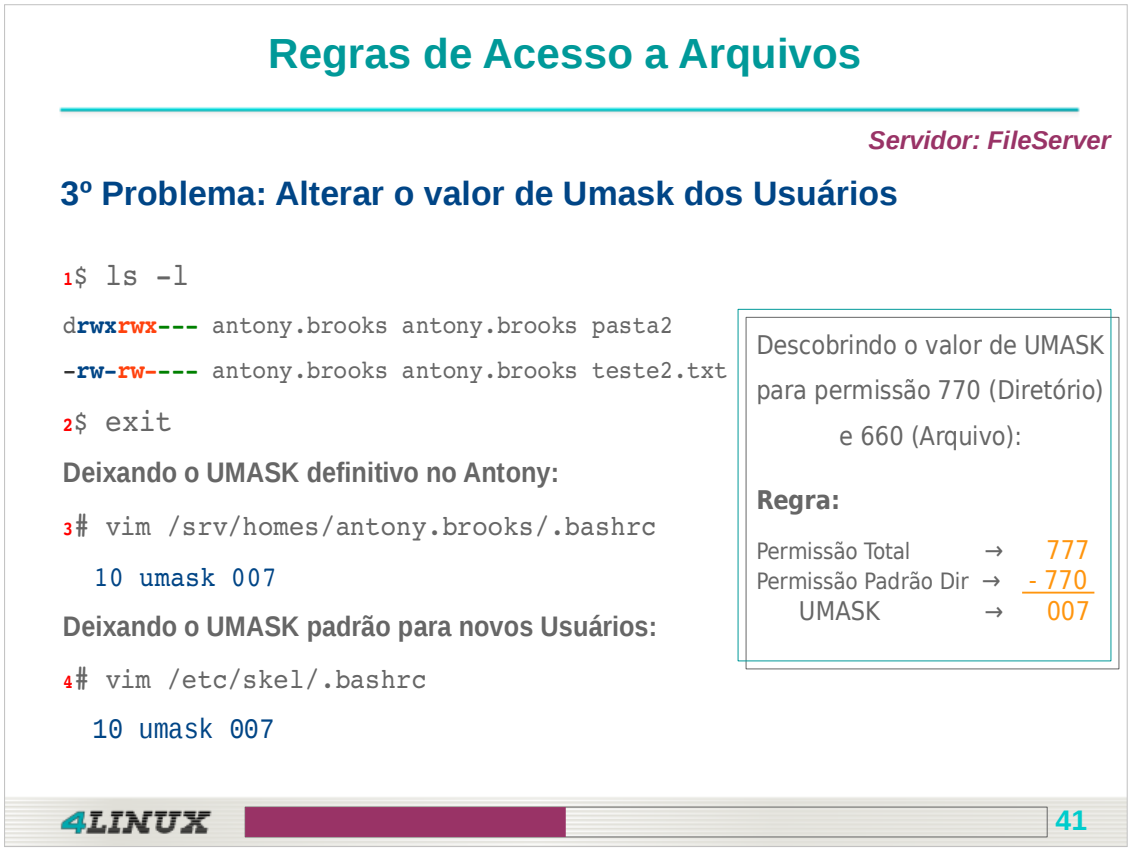
Para calcular a “umask” para um diretório, pegue a permissão total que um diretório pode chegar, “777”. Subtraia “a sua umask atual”.

Exemplo: Considere um umask com valor 022

Primeiro pegamos a permissão máxima para um diretório que é **777** e subtraimos do valor da umask atual que é: **022**

Logo:

777 – 022 = 755 → Este valor é a permissão do diretório a ser criado;



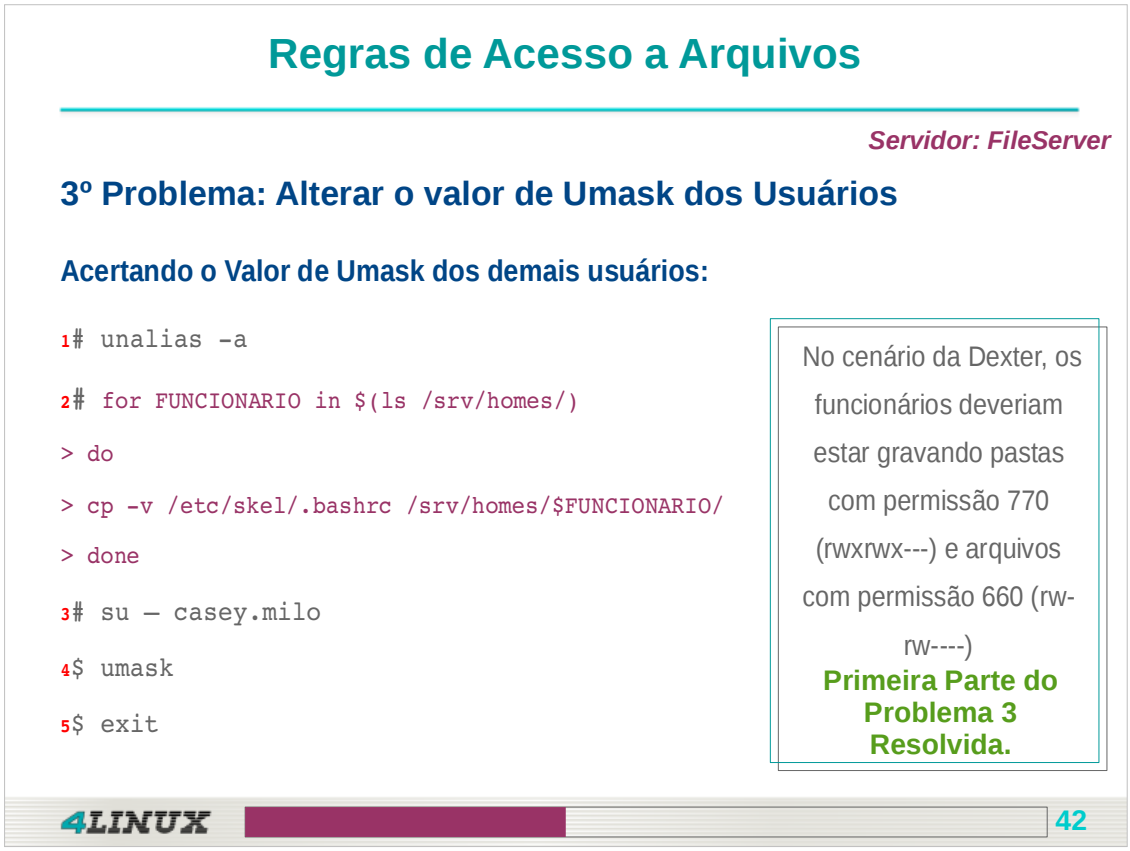
Calculo de umask de arquivos:

Para calcular a “umask” para um arquivo, saiba que **um arquivo não pode ser criado com permissão de execução**, por padrão, esta permissão só pode ser passada para ele manualmente.

Logo a permissão do arquivo não pode ser ímpar, porque o bit de execução vale 1.

Exemplo: Considere a mesma umask do exemplo anterior, 022
Primeiro pegamos a permissão máxima posśivel que é 777 e subtraímos do valor da umask atual que é: **022**

777 – 022 = 755 → Este valor é a permissão calculado mas lembre-se: Não podemos ter permissão de execução sobre os arquivos, sendo o valor da permissão de execução igual a 1, subtraia 1 dos bits que sejam ímpares: **755 – 011 = 644** → Retirando os bits de execução temos 644 que é a permissão real do arquivo.



Resumindo:

Para dominar o cálculo de “umask” basta assumir algumas regras simples:

Para diretórios: Sempre substituir de 777;

Para arquivos: Verificar o “umask”. Se o número for ímpar, subtrair somente onde temos execução, em números pares mantemos os números.

DICA:

Fique atento! Cálculo de Umask costuma ser um arquivo muito abordado em provas LPI.

Regras de Acesso a Arquivos

Servidor: FileServer

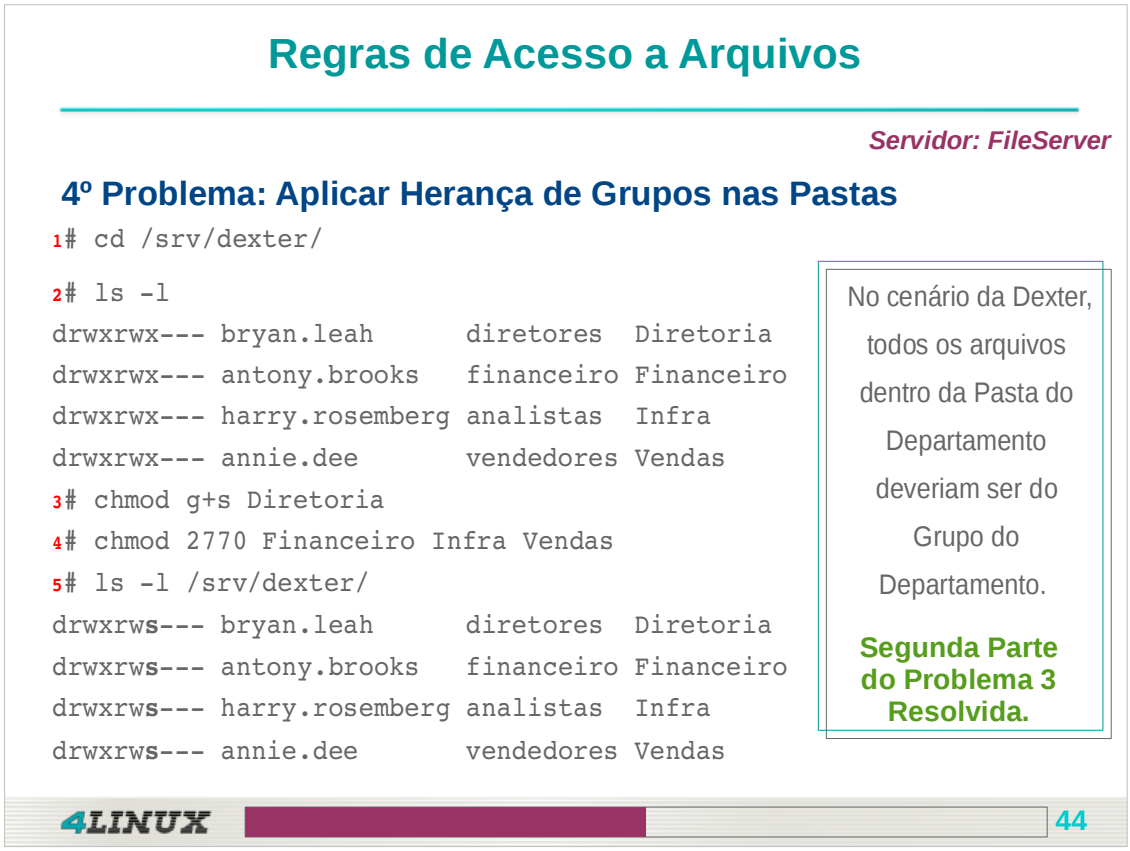
4º Problema: Aplicar Herança de Grupo

Permissões Especiais:

Permissões	Octal	Literal	Descrição
Suid Bit	4	s	Em binários, no momento, da execução herda o “poder” do dono
Sgid Bit	2	s	Em pastas, permite herança de grupos
Stick Bit	1	t	Em pastas, restringe remoção de arquivos e subpastas apenas para o dono

No cenário da Dexter, todos os arquivos dentro da Pasta do Departamento deveriam ser do Grupo do Departamento.

Anotações:



O SGID bit:

O “SGID bit” é geralmente atribuível a diretórios. Quando um arquivo é criado dentro de um diretório com “SGID bit” ativado, o conteúdo gravado dentro do diretório irá herdar o grupo do diretório e não o grupo do usuário que criou tal conteúdo.

Este “bit” especial é muito útil quando utilizamos diretórios para grupos de trabalhos e em servidores de arquivos.

Regras de Acesso a Arquivos

Servidor: FileServer

Validando o Acesso a Pastas e Arquivos:

Financeiro

Funcionários:
Antony Brooks[1], Fox Bennett
Grupo: Financeiro
Pasta Departamental: Financeiro

```
1# rm -rf /srv/dexter/Financeiro/*
2# su – fox.bennett
3$ cd /srv/dexter/Vendas
cd: /srv/dexter/Vendas/: Permissão negada
4$ cd /srv/dexter/Financeiro
5$ touch relatorio.txt
6$ ls -l
-rw-rw----  fox.bennett financeiro  relatorio.txt
7$ exit
```

4LINUX 45

O SGID bit:

O “bit” especial para o campo de permissões do dono é o “SUID” representado por **“s”** ou **“S”**. Para o grupo é “SGID” também representado por **“s”** ou **“S”**.

Veja que quando o arquivo ou diretório não tem permissão de execução, o “bit” especial é representado por uma letra “S” (Upper Case), e quando possuem uma permissão de execução, o “bit” especial é apresentado como “s” (Lower Case). O mesmo acontece com o “Sticky bit” que veremos a seguir.

Regras de Acesso a Arquivos

Servidor: FileServer

Validando o Acesso a Pastas e Arquivos:

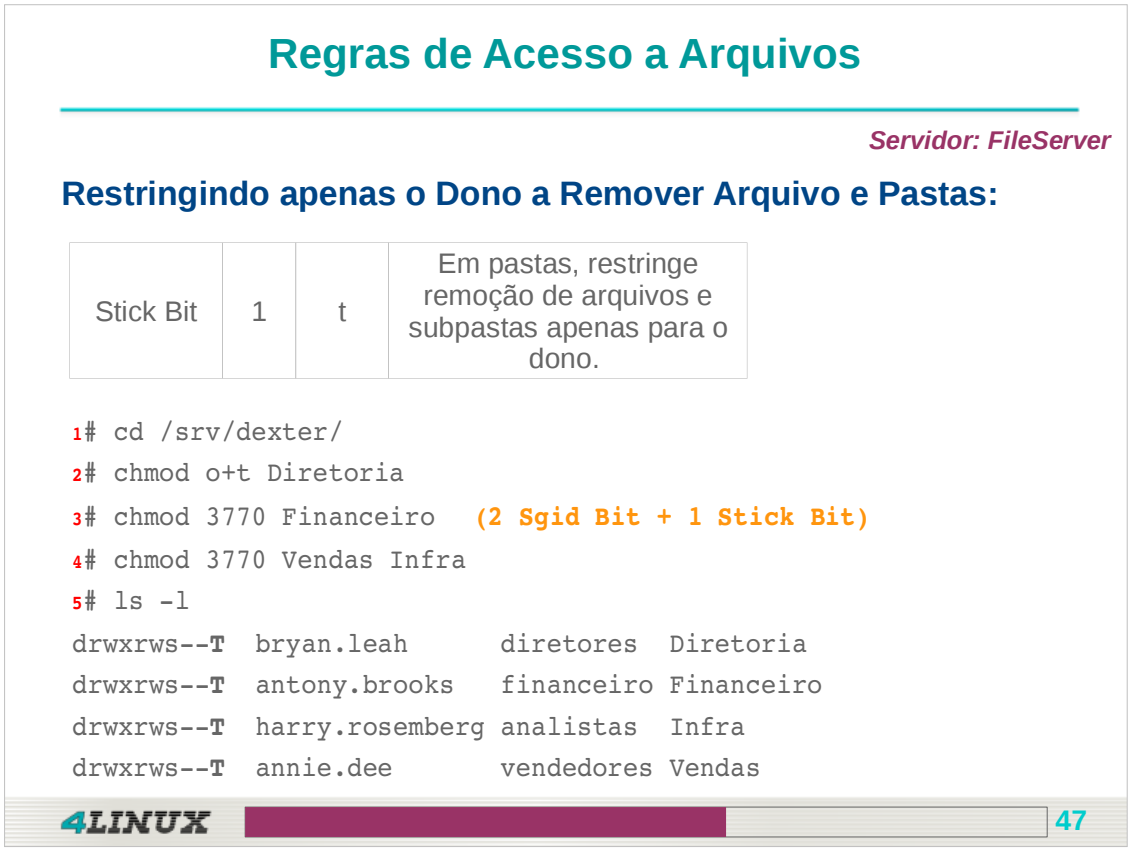
```
1# su – antony.brooks
2$ cd /srv/dexter/Financeiro
3$ echo “Revisão” >> relatorio.txt
4$ cat relatório.txt
5$ mkdir 2013
6$ ls -l
drwxrws---  antony.brooks financeiro  2013
7$ exit
```

Financeiro

Funcionários:
Antony Brooks[1], Fox Bennett
Grupo: Financeiro
Pasta Departamental: Financeiro

3° Problema - OK

Anotações:



Stick bit:

O “Sticky bit” era bastante utilizado para realizar otimizações de acesso a conteúdos, entretanto, a partir da série 2.6 do kernel do Linux essa tarefa é realizada diretamente pelo kernel. A única utilidade desse “bit”, atualmente, é fazer diretórios de utilização comum a todos os usuários, como no “/tmp”.

Quando esse “bit” está ativo em um diretório, todo conteúdo criado dentro dele pertencerá ao criador do conteúdo e por mais que ele atribua a esse conteúdo permissões totais para todos os usuários, o único que poderá excluir o arquivo ou diretório será o próprio dono ou o “root” ou ainda o dono do diretório que tem a permissão.

Regras de Acesso a Arquivos

Servidor: FileServer

Validando Permissão Stick Bit:

Financeiro

Funcionários:
Antony Brooks[1], Fox Bennett
Grupo: Financeiro
Pasta Departamental: Financeiro

```
1# su – fox.bennett
2$ cd /srv/dexter/Financeiro
3$ ls -l
drwxrws--- antony.brooks financeiro 2013
-rw-rw---- fox.bennett   financeiro relatorio.txt
4$ rm -rf 2013
rm: não foi possível remover "2013": Operação não permitida
5$ exit
```

Tente fazer ao contrário: antony (Gerente) apagar o relatório.txt do Fox.
Ele conseguirá pois é dono da pasta, ou seja, gerente pode!

4LINUX 48

Stick bit:

Para atribuirmos esses “bits” especiais, procedemos da mesma forma que nas permissões comuns, somando os valores e utilizando o comando “chmod”, mas agora utilizando quatro números, o primeiro número sendo o “bit” especial, seguido dos três da permissão padrão.

O “Sticky BIT”, é representado por **“t”** ou **“T”**.

Cenário Concluído!

O Servidor FileServer está pronto para receber o Samba 4 e se tornar o PDC/FileServer da Dexter! (Curso 4452)

Anotações:

Laboratório Dexter

Servidor: FileServer

O Presidente Dexter Clem acessa todas as pastas dos Departamentos:

```
1# su – dexter
2$ cd /srv/dexter/Financeiro
cd: /srv/dexter/Financeiro/: Permissão negada
3$ exit
```

Sem usar ACL do Linux para resolver essa última exigência, o Presidente deverá pertencer ao grupo de todos os Departamentos!

Anotações:

Laboratório Dexter

Servidor: FileServer

O Presidente Dexter Clem acessa todas as pastas dos Departamentos:

```
1# gpasswd -a dexter diretores
2# gpasswd -a dexter financeiro
3# gpasswd -a dexter vendedores
4# gpasswd -a dexter analistas
5# groups dexter
6# su – dexter
7$ cd /srv/dexter/Financeiro
8$ groups
9$ exit
```

```
dexter : dexter diretores vendedores financeiro analistas
```


Anotações:

Regras de Acesso a Arquivos

Servidor: Audit

Exemplo de SUID Bit na máquina Audit:

Suid Bit	4	s	Em binários, no momento da execução, herda o “poder” do dono

```
1# su – suporte
2$ ifconfig
ifconfig: comando não encontrado
3$ /sbin/ifconfig
4$ /sbin/ifconfig eth0:1 10.0.0.1
SIOCSIFFLAGS: Permissão negada
5$ exit
6# chmod u+s /sbin/ifconfig
```

```
7# su – suporte
8$ ifconfig
ifconfig: comando não encontrado
9$ /sbin/ifconfig
10$ /sbin/ifconfig eth0:1 10.0.0.1
11$ /sbin/ifconfig
12$ exit
13# chmod u-s /sbin/ifconfig
14# ifconfig eth0:1 down
15# service networking restart
```

4LINUX 52

Suid bit:

O “SUID bit” é atribuído a um arquivo binário com permissão de execução, quando desejamos que um usuário qualquer execute o comando com as permissões do usuário dono do comando.

Se esse comando pertencer ao usuário “root” um usuário qualquer irá executá-lo com as permissões de “root” desde que tenha permissões para executá-lo.

Por esse motivo o “SUID” constitui uma grande ameaça de segurança e sua utilização deve ser bastante cautelosa.

Pergunta LPI

Selecione a melhor opção que representa as permissões de acesso ao arquivo /etc/passwd?

A. `rw-r--r-- 1 1 1 531 Jun 5 22:45 /etc/passwd`

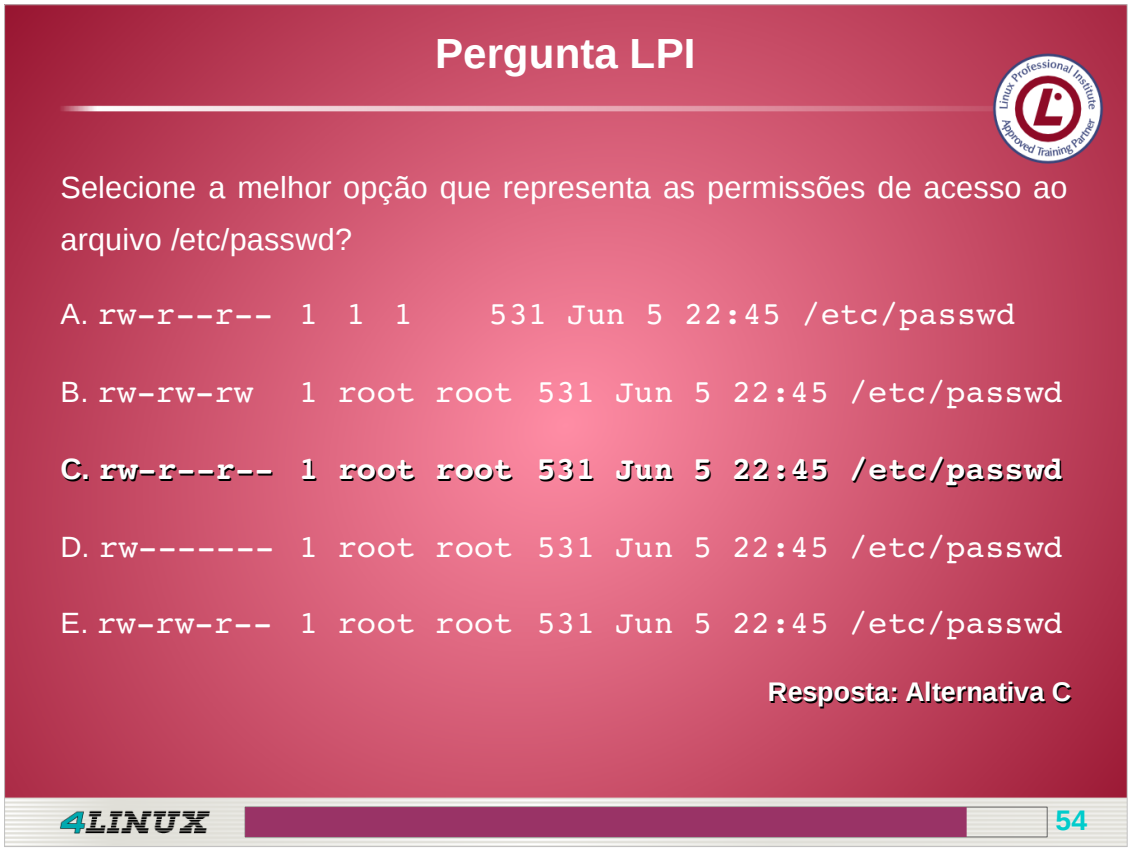
B. `rw-rw-rw 1 root root 531 Jun 5 22:45 /etc/passwd`

C. `rw-r--r-- 1 root root 531 Jun 5 22:45 /etc/passwd`

D. `rw------- 1 root root 531 Jun 5 22:45 /etc/passwd`

E. `rw-rw-r-- 1 root root 531 Jun 5 22:45 /etc/passwd`

Anotações:



Alternativa C: RESPOSTA CORRETA!

O arquivo **passwd** é utilizado nas operações de autenticação de usuários no sistema, logo é preciso que qualquer nível de usuário possua permissão de leitura sobre o arquivo, por este motivo, as senhas de usuários ficam armazenadas no arquivo /etc/shadow, sendo este, restrito ao usuário root como permissões rw-------.

Pergunta LPI

O diretório /etc/skel é utilizado para:

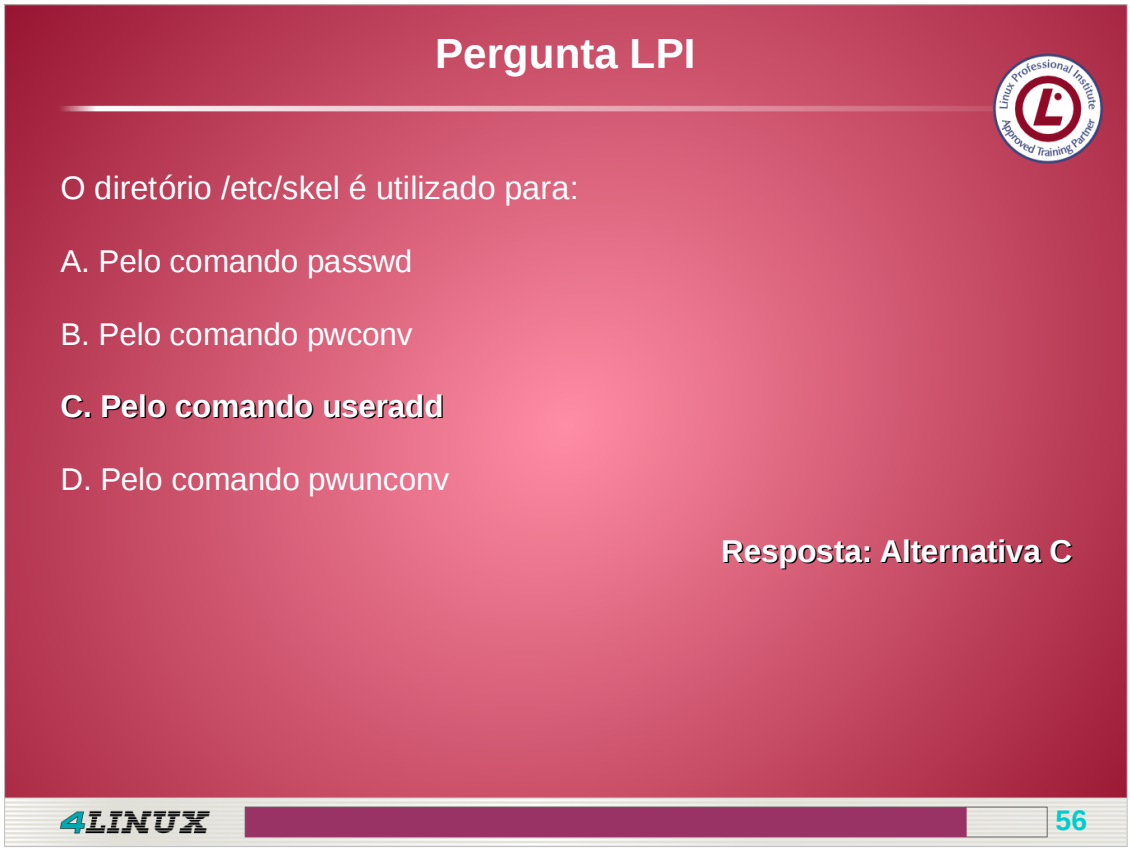
A. Pelo comando passwd

B. Pelo comando pwconv

C. Pelo comando useradd

D. Pelo comando pwunconv

Anotações:



Alternativa C: REPOSTA CORRETA!

Ao criar um usuário através do comando **useradd** ele fará uma cópia do conjunto de pastas e arquivos definidos no diretório **/etc/skel** para gerando todo o escopo do diretório home do usuário recém criado.

Próximos Passos

Para que você tenha um melhor aproveitamento do curso, participes das seguintes atividades disponíveis no Netclass:

- Executar as tarefas do **Practice Lab**;
- Resolver o **Desafio Appliance Lab** e postar o resultado no Fórum Temático;
- Responder as questões do **Teste de Conhecimento** sobre o conteúdo visto em aula.

Mãos à obra!