Otázka 29 - Y38UOS

Zadání

Identita uživatelů, procesů a souborů v OS Unix. Přístupová práva a jejich nastavení. (Y38UOS)

Slovníček pojmů

Identita uživatelů, procesů a souborů v OS Unix

Identita uživatelů

Je tvořena přihlašovacím jménem a heslem. Slouží k přihlášení do systému, pojmenováni domovského adresáře.
Při přihlášení do systému musí uživatel

- **identifikovat systém, na který se chce přihlásit(fyzické umístění(lokální přihlášení), jméno systému/IP adresa(vzdálené přihlášení))**
- **zadat uživatelské jméno**
- **prokázat se odpovídajícím heslem**

Pro úspěšné přihlášení musí být na daném systému vytvořen příslušný uživatelský účet.

Přihlášení do systému

- **systém vypíše na příslušném zařízení prompt**
- **uživatel vloží uživatelské jméno a odpovídající heslo**
- **systém ověří vložené informace proti databázi uživatelských účtů**
- **systém spustí přihlašovací shell a nastaví pro tento proces:**

```
pracovní adresář na domovský adresář daného účtu
reálné číslo uživatele RUID = UID
efektivní číslo uživatele EUID = UID
reálné číslo primární skupiny RGID = GID
efektivní číslo primární skupiny EGID = GID
```

Uživatelský účet

Je vytvořeny správcem (administrátorem) každému uživateli. Účet jednoznačně definuje uživatele a jeho pracovní prostředí. Ve většině případů se účty zapisují do souboru /etc/passwd. V tomto souboru byla původně uložena i hesla, ale kvůli bezpečnosti byla přemístěna do souboru /etc/shadow. Tento soubor je zašifrovaný a šifra je jednocestná.

Uživatelský účet obsahuje tyto položky:

uživatelské jméno (jméno)
Uživatelské jméno označuje konkrétního uživatele operačního systému a je většinou tvořeno skupinou 3 - 8 znaků. Takto se tedy uživatel hlásí do systému. Jméno musí vždy začínat písmenem a vždy se zapisuje malými písmeny. Uživatelské jméno může obsahovat i číslice.

identifikační číslo uživatele (UID)
Jedná se o číslo uživatele, které Unix používá k identifikaci namísto uživatelského jména. Je to výhodnější, protože je rychlejší najít uživatele podle určitého UID a ten potom předávat různým

programům, než porovnávat jestli neexistuje jiný uživatel s podobně začínajícím loginem. Používá se uvnitř datových struktur. Při komunikaci s uživatelem se přes soubor /etc/passwd překládá do textové podoby uživatelského jména. Přidělováno administrátorem.UID=0 definuje tzv. privilegovaný účet (obvykle se jménem root)

identifikační číslo primární skupiny (GID)
Každý uživatel může být ve více skupinách, ale primární skupinou je ta, která je uvedená v souboru /etc/passwd. Členství v této skupině již nemusí být zaznamenáno v souboru /etc/group. Soubor /etc/group obsahuje seznam skupin včetně přiřazení uživatelů do těchto skupin. Jednotlivé položky na řádce jsou odděleny dvojtečkou. U každé skupiny jsou uvedeny tyto údaje: jméno skupiny:heslo:GID:eventuelně seznam členů. Uživatel se může přepínat z jedné skupiny do druhé a to za použití příkazu newgrp.

přihlašovací interpret (shell)

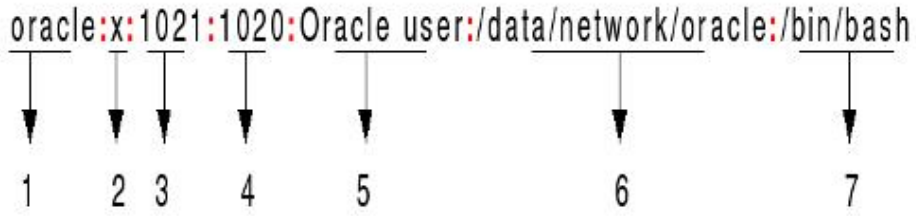
- **absolutní cesta k příkazu, který je spuštěn po přihlášení uživatele z řádkového terminálu**
- **obvykle shell, ale může být i jiný program**

domovský adresář
Je to absolutní cesta k adresáři, která se po přihlášení nastaví jako běžný adresář a který je nadále označován jako domovský adresář uživatele. Je to adresář, ze kterého je spuštěn přihlašovací interpret a je na něj nastavena proměnná HOME.

Heslo
Tak jako v jiných operačních systémech, kde se používá heslo, tak i zde v Unixu si heslo volí uživatel. Toto heslo se v počítači ukládá do souboru /etc/passwd nebo /etc/shadow. Heslo je do těchto souborů vkládáno v zakódované (hašované) podobě. Druhý soubor, tedy /etc/shadow je pro ukládání hesel vhodnější, jelikož se zde nastaví právo pouze pro čtení, a to pouze pro správce systému a nikdo jiný se tedy k heslům nedostane. Hesla pro skupiny se ukládají do souboru /etc/gshadow. Heslo účtu není povinné a uživatelé tedy nejsou povinni si svůj účet chránit heslem. Pokud má však k počítači přístup více než jeden uživatel, je velice vhodné si heslo vytvořit. Toto heslo by pak mělo mít alespoň 6 znaků a nemělo by se skládat pouze z písmen, ale vhodné je použít i číslice a další znaky.

/etc/passwd

Primárně slouží k překladu UID na jméno a naopak a k uložení informací nutných pro přihlášení uživatele. Pro každý uživatelský účet je jedna řádka rozdělená na sedm položek



- **1 - Přihlašovací jméno: Používá se při přihlášení uživatele. Mělo by být mezi 1 až 32 znaků.**
- **2 - Heslo: x znak znamená, že zašifrované heslo je uloženo v /etc/shadow soubor.**
- **3 - ID uživatele (UID): Každý uživatel musí mít jedinečné ID(UID). UID 0 (nula) je vyhrazena pro root.**
- **4 - ID skupiny (GID): ID skupiny (v souboru / etc/group file)**
- **5 - Uživatel ID Info: komentář pole. To vám umožní přidat další informace o uživatelích, jako je uživatelské jméno a příjmení, telefonní číslo atd.**
- **6 - Domovský adresář: absolutní cesta home adresáře uživatele.**
- **7 - Command / shell: absolutní cesta výchoziho shell.**

/etc/group

Slouží k překladu GID na jméno skupiny a naopak a k definici tzv. sekundárních skupin. Pro každou skupinu jedna řádka rozdělena znaky : na čtyři pole.

```
jméno skupiny:x:GID:seznam uživatelů
```

/etc/shadow

Obsahuje zašifrované heslo a parametry nastavení hesla pro každý uživatelský účet jedna řádka rozdělená znaky : na devět položek

```
jméno:heslo:lastchg:min:max:warn:inactive:expire:flag
```

Identita procesů

Tvoří ji identifikační a efektivní identifikační číslo uživatele a skupiny (uid, gid, euid, egid). Jsou to celá čísla sloužící k autorizaci procesu uvnitř systému (ověření práv k provádění operací se soubory a práv ke spouštění jiných procesů).

Reálná identita procesu (RUID, RGID)

- **systém si pamatuje pod jakým uživatelským účtem jsme se původně přihlásili (např. lokálně nebo vzdáleně pomocí ssh,...)**
- **lze ji zobrazit např. pomocí následujících příkazů who am i , ps -o ruid,rgid,comm**

Efektivní (aktuální) identita procesu (EUID, EGID)

- **slouží k autorizaci procesu uvnitř systému (např. při vyhodnocování přístupu k souborům, ...)**
- **při přihlášení je reálná a efektivní identita totožná**
- **efektivní identitu lze změnit např. pomocí příkazu su nebo speciálních práv binárních souborů**
- **lze ji zobrazit např. pomocí následujících příkazů id, ps –o uid,gid,comm**

Změna identity procesu

Identitu procesu nastavuje kernel při startu procesu nebo ji mění na žádost procesu. Obvykle jsou RUID a EUID resp. RGID a EGID stejná a dědí se od rodičovského procesu. Ve zvláštních případech se nedědí, ale nastavují se všechna nebo jen některá ID:

- **při přihlášení (pomocí procesů login/dtlogin)**
- **pomocí příkazu su**
- **u binárních programů s nastaveným suid bitem se mění EUID**
- **u binárních programů s nastaveným sgid bitem se mění EGID**

Příkaz su

su [-] [uživatelské jméno]

- **startuje nový shell pod novou identitou**
- **původní shell nekončí, po odhlášení ze su se v něm pokračuje**
- **je-li su volán uživatelem, vyžaduje heslo, od roota ne**
- **je-li uveden přepínač -, provede přihlašovací skripty (nastaví prostředí)**
- **je-li vynecháno přihlašovací jméno, doplní se jméno root**

Identita souborů

Soubor je posloupnost bytů identifikovaná jménem.

Typy souborů

- **obyčejné soubory (textové soubory, binární soubory)**
- **adresářové soubory (čili adresáře)**
- **speciální soubory (reprezentují rozhraní mezi zařízením a OS)**
- **symbolické linky (soubory, které obsahují odkaz na jiný soubor)**
- **pojmenované roury (named pipes - mechanismus spojování procesů do řetězce pomocí vzájemného propojení standardních proudů tak, že výstup stdout procesu je nasměrován do vstupu stdin následujícího procesu)**
- **sockety (prostředek meziprocesní komunikace, kdy komunikující procesy nemusí být na stejném počítači)**

Jména souborů a adresářů

- **maximální délka závisí na implementaci, obvykle 255 znaků**
- **nepovolené znaky: lomítko /**
- **nedoporučené znaky: * ? [ ] ( ) \ “ “ ‘ ‘ ` ` + ;**
- **doporučené znaky: alfanumerické znaky, tečky, pomlčka (ne na začátku), podtržítko**
- **rozlišování velkých a malých písmen**
- **jména začínající tečkou jsou skryté soubory/adresáře**
- **rezervovaná jména: tečka (.) aktuální adresář a dvě tečky (..) nadřazený adresář**

Každý soubor má svého vlastníka. Vlastníkem souboru je automaticky ten, kdo jej vytvořil. Vlastník definuje přístupová práva pro ostatní uživatele:
– patřící do stejné skupiny uživatelů jako vlastník
– nepatřící do stejné skupiny jako vlastník

Root může měnit přístupová práva ke všem souborům a může rovněž měnit vlastníka souboru. Identifikace vlastníka je dle UID, identifikace skupiny je dle GID.

Přístupová práva a jejich nastavení

Přístupová práva v Unixu umožňují ve víceuživatelském systému definovat přístup k adresářům a souborům na základě uživatelských účtů nebo skupin uživatelů. Kontrola přístupu umožňuje na systémové úrovni zabránit uživatelům, aby záměrně nebo omylem cizí data poškodili nebo zneužili.

Přístupová práva Jsou to atributy každého souboru i složky, které určují, jaká práva nad tímto souborem/složkou mají určití uživatelé (skupiny). Tyto atributy se zapisují přímo do tabulky I-nodů.

Každý soubor/adresář má v i-uzlu

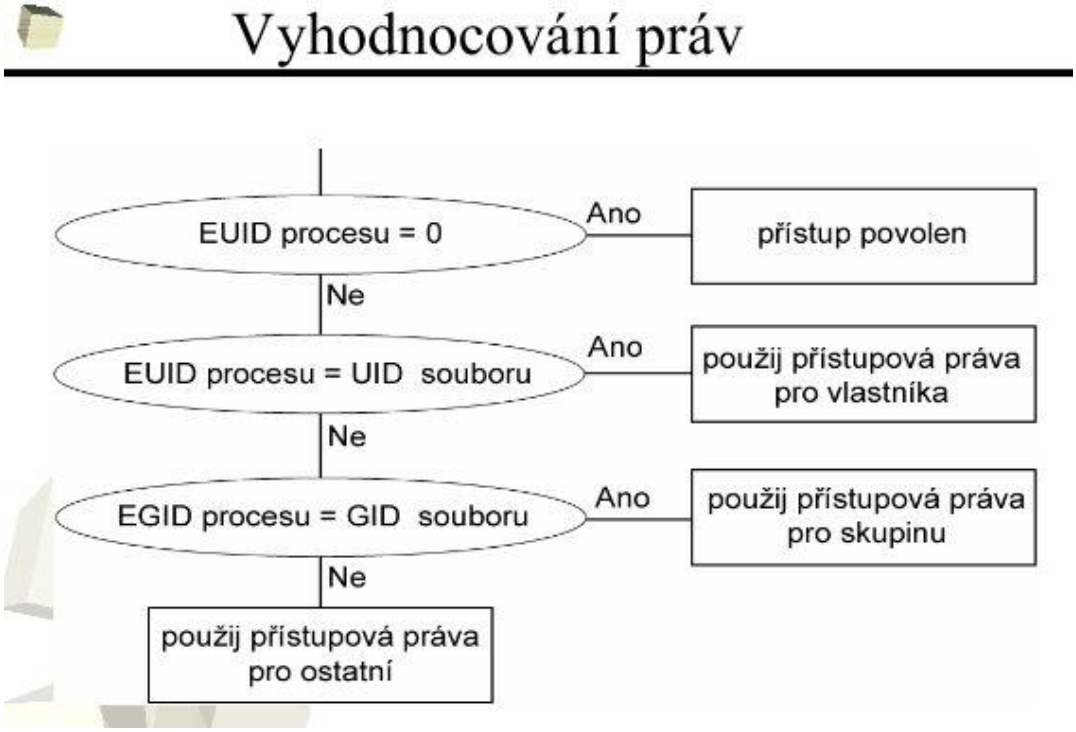
- **vlastníka souboru (UID)**
- **vlastnickou skupinu (GID)**
- **přístupová práva čtení (read), zápis (write) a spuštění (execute) pro vlastníka (user), skupinu do které vlastník patří(group), pro ostatní (other) a pro všechny (all).**

Tyto informace můžeme vypsat např. pomocí příkazu ls –l

První pomlčka znamená, že se jedná o obyčejný soubor (není to adresář d , ani symbolický odkaz l, apod.) Následují pak tři skupiny rx za sebou. Pokud má soubor nějaké právo nastaveno, vídíme odpovídající písmenko, pokud ne, vídíme jen pomlčku. Skupiny rx jsou tři, protože vyjadřují tři typy uživatelů. V dalších polích je uloženo:

- **počet odkazů na soubor (tj. na tentýž i-uzel)**
- **vlastník souboru**
- **skupina do které vlastník patří**
- **velikost souboru v bytech**
- **datum vytvoření souboru**
- **jméno souboru**

V uvedeném příkladě může tedy vlastník, skupina do které vlastník patří a ostatní číst a spouštět soubor.



Zápis oprávnění

V unixových systémech se práva zapisují oktalově nebo pomocí symbolického zápisu (binární zápis se nepoužívá). Výsledná hodnota je součtem hodnot jednotlivých oprávnění v každé trojici

Typ práva	Symbolické vyjádření	Oktalové vyjádření
Čtení	r (Read)	4
Zápis	w (Write)	2
Spuštění	x (eXecute)	1

Význam oprávnění

Oprávnění pro soubory a adresáře se významově poněkud liší, jak zachycuje následující tabulka:

	Soubor	Adresář
Read	čtení ze souboru	čtení adresáře (výpis obsahu)
Write	zápis do souboru (změna obsahu, délky)	zápis do adresáře (vytváření, mazání a přejmenov
eXecute	spuštění (program, skript)	vstup do adresáře

Příklady zápisu oprávnění

700

Vlastník	Skupina	Ostatní
r w x	r w x	r w x
4 2 1	0 0 0	0 0 0

600

Vlastník	Skupina	Ostatní
r w x	r w x	r w x
4 2 0	0 0 0	0 0 0

755

Vlastník	Skupina	Ostatní
r w x	r w x	r w x
4 2 1	4 0 1	4 0 1

644

Vlastník	Skupina	Ostatní
r w x	r w x	r w x
4 2 0	4 0 0	4 0 0

- **700 Vlastník souboru může číst, zapisovát do souboru. U adresáře může zápisovat a otevírat adresář.**
- **600 Vlastník souboru může zapisovat a číst tento soubor.**
- **755 tabulka právo čtení a otevření adresáře mají všichni, měnit data může vlastník adresáře.**
- **644 právo čtení souboru mají všichni, měnit data může jen vlastník souboru.**

Změna přístupových práv

Na změnu práv slouží príkaz chmod. Měnit práva souboru, adresářu může buď vlastník a nebo root

```
chmod [-R] práva seznam_souborů
```

-R(recursive) změna práv se aplikuje na všechny soubory a podadresáře v daném adresáři.

Měnit práva můžeme buď pomoci symbolického zápisu

```
chmod u+x,g-r,o+w a.txt
```

Plus znamená přidat právo, mínus znamená odebrat právo, rovnítko znamená nastav práva (tj. existující práva nahraď).

Nebo absolutně(oktalově)

```
chmod 706 a.txt
chmod 700 muj_supertajny_soubor
```

Změna vlastnictví souboru

Může měnit pouze root
Vlastnictví (i skupinové) lze měnit příkazem chown, skupinové příkazem chgrp.

```
chown [-R] vlastník [:skupina] seznam_souborů
chgrp [-R] skupina seznam_souborů
```

Maska přístupových práv

- **Definuje přístupová práva nově zakládaných souborů/adresářů.**
- **Hodnota masky je součástí procesu (podobně jako EUID,EGID,...) a je dědičná.**
- **Lze ji vypsat a měnit příkazem umask.**
- **Přístupová práva vzniknou množinovým rozdílem výchozí hodnoty a masky.**
- **Výchozí hodnota je 666 pro soubory a 777 pro adresáře.**

maska	soubor	adresář	poznámka
000	666	777	odpovídá výchozí hodnotě, Nebezpečné
022	644	755	obvyklé nastavení
027	640	750	vyšší bezpečnost
077	600	700	největší restrikce
066	600	711	kompromisní řešení

Platí: maska + soubor = výchozí hodnota soubor, maska + adresář = výchozí hodnota adresář.

Speciální oprávnění

Speciální oprávnění mění standardní chování systému, což je výhodné v některých speciálních případech.

SUID (setuid)

Za standardních okolností dědí potomek (nový proces) oprávnění svého rodiče. Někdy je však nutné, aby měl spuštěný program jiná (vyšší) oprávnění. Pokud je na souboru s programem nastaven SUID bit, neběží spuštěný program s právy rodiče, ale s právy vlastníka tohoto souboru. Používá se v případech, kdy chceme uživateli umožnit provedení akce, na které by potřeboval jiná nebo vyšší oprávnění:

Změna hesla
Hesla uživatelů jsou uložena v souboru /etc/passwd nebo /etc/shadow, do kterých běžný uživatel nemůže zapisovat (soubor shadow nemůže dokonce ani číst). Při změně hesla je ale potřeba změněné heslo do těchto souborů zapsat. Proto má program /usr/bin/passwd nastaven SUID bit a patří uživateli root. Po spuštění běží program passwd s právy roota a heslo může být do příslušného souboru zapsáno.

Změna uživatele
Program běžného uživatele nemůže změnit svoje oprávnění. Může to však udělat program běžící s právy roota. Proto má program su nastavený SUID bit a patří uživateli root'. Po zadání správného hesla je spuštěn nový shell, který je nastaven na nová oprávnění.
SUID bit neznamená, že program poběží s právy uživatele root. Patří-li program jinému uživateli,

bude po spuštění běžet s právy tohoto uživatele. Nejběžnější je však tento způsob při poskytování administrátorských oprávnění. Proto musí být každý program se SUID bitem naprogramován s maximální obezřetností, aby neumožnil provést nějakou neoprávněnou činnost.

SGID (setgid)

Program s nastaveným SGID bitem se chová po spuštění podobně, jako u SUID bitu. Nepřebírá ale oprávnění majitele souboru, nýbrž oprávnění skupiny, které daný soubor s programem na disku patří. Při aplikaci SGID bitu na adresář patří všechny nově vytvořené soubory a adresáře do skupiny, která je shodná s nadřízeným adresářem (který má nastaven zmíněný SGID bit). Bez nastaveného SGID bitu patří nově vytvořené adresáře a soubory primární skupině uživatele (viz výše).

Skóre ve hře
Některé hry zapisují dosažené skóre do souboru, aby mohli hráči své výkony porovnat. Takový soubor by musel mít právo zápisy pro všechny uživatele v systému. Hráči by pak snadno mohli tento soubor měnit a své dosažené skóre neférově zvyšovat. Proto je program s hrou svěřen speciální skupině (např. games) a je mu nastaven SGID bit. Soubor se skóre pak bude mít právo zápisu přidělené jen skupině games. Do souboru se skóre tak spuštěná hra může zapisovat, kdežto uživatelé nemohou soubor měnit.

Skupinový projekt
Uživatelé, kteří pracují na společném projektu patří do společné skupiny project a obvykle nemají tuto skupinu nastavenou jako primární (nebo si ji zapomenou před každou prací na projektu pomocí příkazu newgrp změnit). Vytvoří-li ve společném adresáři, kam mají na základě členství ve skupině project přístup, nový soubor nebo adresář, bude patřit jiné skupině. Ani při nastavení umask na hodnotu zajišťující skupině zápis tak nebudou kolegové moci soubory upravovat nebo soubory v nových adresářích mazat a přejmenovávat. Přidělit oprávnění všem není nikdy vhodné. Proto je na kořenový adresář projektu, který patří skupině project nastaven SGID bit. Nové soubory a adresáře tak automaticky patří skupině project a nové podadresáře mají nastaven SGID bit.

Sticky bit

Pokud má uživatel do adresáře právo zápisu, může v tomto adresáři i mazat. Nastavíme-li v adresáři sticky bit, bude uživatel smět smazat jen své vlastní soubory nebo adresáře. Ve starších verzích unixových systémů tento příznak na souborech sloužil k tomu, aby kód ukončeného programu zůstal ve swapu. Tímto způsobem bylo dosaženo rychlejšího startu často používaných programů. V dnešních systémech není sticky bit na souborech již delší dobu podporován, protože vlivem používání virtuální paměti a stránkování ztratil pro soubory význam.

Adresář /tmp
Do adresáře /tmp mají všichni uživatelé v systému právo zápisu a odkládají si tam dočasné soubory (zejména to dělají různé programy). Pokud mají právo zápisu, mohou v adresáři i přejmenovávat a mazat soubory i adresáře, což by vedlo k tomu, že by si uživatelé mohli navzájem škodit. Nastavíme-li na tento adresář sticky bit a uživatelé budou vytvářet své soubory a adresáře s adekvátními oprávněními, bude problém vyřešen.

Zdroje

Slajdy předmětu Y36UOS
http://blog.milde.cz/post/88-pristupova-prava-v-unixu/ [http://blog.milde.cz/post/88-pristupova-prava-v-unixu/]
http://vse.ugic.net/doc/4IT425_Prednaska_I-3.pdf [http://vse.ugic.net/doc/4IT425_Prednaska_I-3.pdf]

spolecne/spol29.txt · Poslední úprava: 2010/06/13 22:48 autor: Petronyuk