Użytkownicy I

Głównym celem istnienia użytkowników i grup w systemie jest utrzymanie porządku i separacja uprawnień.

Użytkownik

- login (nazwa)
- UID – identyfikator numeryczny
- przynależność do grup, w tym dokładnie jedna grupa podstawowa
- hasło
- katalog domowy
- powłoka
- ...

Użytkownicy II

Grupa

- GID – identyfikator
- Nazwa
- Lista członków

Użytkownicy III

Lokalne konta uzytkowników — plik /etc/passwd

- `man 5 passwd`
- Linie postaci: `login:password:uid:gid:gecos:home:shell`
- Dawniej (20+ lat) hasło przetworzone trudno odwracalną funkcją było w publicznie dostępnym pliku passwd. Obecnie hasła są w odrębnym pliku dostępnym tylko dla administratora, np. GNU/Linux: `/etc/shadow`, FreeBSD: `/etc/master.passwd`,
- home – ścieżka do katalogu domowego
- shell – powłoka używana przez użytkownika (w laboratorium dostępny tylko bash)
- gecos – informacje o użytkowniku, np. imię, nazwisko, numer telefonu, numer pokoju.

Użytkownicy IV

Lokalne grupy – /etc/group

- `man 5 group`
- Linie postaci: `name:password:gid:members`
- Hasło – w niektórych systemach istnieje możliwość (bardzo rzadko używana) czasowego pozyskania przynależności do grupy po podaniu dodatkowego hasła.
- Członkowie grupy: $M_g \cup M_p$, gdzie M_g – członkowie wymienieni w /etc/group, M_p – użytkownicy, którzy mają GID tej grupy jako grupę podstawową.

Użytkownicy V

Inne źródła informacji o użytkownikach – NSS Switch (typowe w Linuksie)

- Plik `/etc/nsswitch.conf`, `man nsswitch.conf`
- Definiuje z jakich źródeł mają pochodzić informacje z poszczególnych baz (passwd, group, itp.)
- Możliwość podłączenia bazy użytkowników na centralnym serwerze
- Program `getent` pozwala odpytywać wszystkie bazy
- `grep karwowskij /etc/passwd` – nie ma takiego wpisu
- `getent passwd karwowskij`

Użytkownicy VI

PAM (Typowe w linuksie)

Pluggable Authentication Modules

- Możliwość zdefiniowania innych metod weryfikacji użytkownika niż hasło
- Wiele modułów, np. tokeny jednorazowe, odciski palców
- Możliwość dodawania różnych ograniczeń, np. logowanie w określonych godzinach.
- Katalogi `/etc/security/`, `/etc/pam.d/`

Użytkownicy VII

Modyfikacja danych przez użytkowników

`$passwd` – zmiana hasła
`$chfn` – zmiana informacji w polu gecos
`$chsh` – zmiana domyślnej powłoki

root (administrator systemu) może wywołać w/w polecenia z parametrem – nazwą użytkownika, którego dane chce zmienić.

Rodzaje uzytkowników – typowa konwencja

- UID$=0$ — root, administrator systemu
- UID$\in [1, 1000)$ (czasami $[1, 500)$) – konta wykorzystywane przez daemony (usługi systemowe)
- UID$\geqslant 1000$ (500) — konta zwykłych uzytkowników

Użytkownicy VIII

su, sudo – programy do uzyskania uprawnień innego użytkownika

W większości systemów uniksowych można spotkać programy `sudo` i `su`, które służą do uzyskania uprawnień innego użytkownika na zasadach określonych przez administratora.

Administracja kontami uzytkowników

POSIX nie definiuje tego elementu w żaden sposób.
W większości systemów występują programy wspomagające zarządzanie użytkownikami, o nazwach podobnych do `useradd`, `adduser`, `usermod`.
Programy są zróżnicowane między systemami, w tym miedzy dystrybucjami Linuksa.
Zazwyczaj występuje katalog `/etc/skel`, który zawiera pliki kopiowane do katalogu domowego nowo tworzonego użytkownika.

Uprawnienia do plików I

Właściciele

Każdy plik ma:

- Właściciela (użytkownika) – zazwyczaj jest nim twórca pliku
- Grupę właścicieli – zazwyczaj podstawowa grupa twórcy pliku

Uprawnienia do plików II

Podstawowe bity uprawnień

u – właścicel, g – grupa, o – pozostali, r – odczyt, w – zapis, x – wykonanie. x oznacza, że dany plik można uruchomić jako program.

u			g			o		
r	w	x	r	w	x	r	w	x
1	1	0	1	0	0	0	0	0

Przykładowe uprawnienia: właściciel odczyt-zapis, grupa odczyt, pozostali brak uprawnień.

Posługujemy się dwoma rodzajami kodowania uprawnień: symbolicznym i numerycznym.

Uprawnienia do plików III

Zapis uprawnień – symboliczny

Dziewięć pozycji znakowych: `rwxrwxrwx`. Pierwsza grupa to uprawnienia właściciela, gruda grupy, trzecia – pozostali.
Zapisujemy odpowiednią literę, jeśli przyznano uprawnienie, kreskę w przeciwnym przypadku.
`rw-r-----` — odczyt zapis dla właściciela, odczyt dla grupy

Zapis uprawnień – numeryczny

Mamy dziewięć bitów uprawnień — liczbę binarną. Kolejność bitów jak poprzednio: `rwxrwxrwx`. Można zapisać w systemie pozycyjnym o podstawie 8 (dokładnie trzy bity na cyfrę).
`rw-r-----` $\equiv 110100000_2 = 640_8$
Zapis ósemkowy jest typowym zapisem numerycznym uprawnień. Często, w stylu języka C, poprzedzamy liczbę w ósemkową 0: 0640.
$r = 4$, $w = 2$, $x = 1$

Uprawnienia do plików IV

Zmiana właścicieli

```
# chown [-R] użytkownik:grupa sciezka1 sciezka2...
```

Przypomnienie: # oznacza komendę wywoływaną przez administratora.

```
$ chown [-R] :grupa sciezka1 sciezka2...
```

Zwykły uzytkownik może zmienić grupę pliku, którego jest właścicielem na inną grupę, której jest członkiem.

Uprawnienia do plików V

Zmiana uprawnień

`$ chmod uprawnienia scezka1 sciezka2...`

Uprawnienia mogą być zdefiniowane na wiele sposobów:

- Numerycznie
- Symbolicznie według schematu: `[ugoa]+[+-=][rwx]*` Podajemy: komu zmienić uprawnienia (a – wszyscy, czli ugo), czy zwiększyć względem bieżących, zmniejszyć, ustawić nowe, uprawnienia do odpowiednio przyznania, odebrania, ustawienia.

Stat

`$stat sciezka` – wyświetlenie informacji o pliku, między innymi uprawnień i właścicieli

Uprawnienia do plików VI

Uprawnienia do katalogów

- r — odczyt nazw plików w katalogu
- w — zmiany w liście plików w katalogu (usuwanie, tworzenie plików)
- x — operacje na plikach wewnątrz katalogu (nazywane też wejściem do katalogu), otwarcie pliku wewnątrz katalogu, pobranie danych szczegółowych o tym plikach, wejście do podkatalogów (o ile podkatalog też ma x).

Uprawnienia do plików VII

Uprawnienia specjalne

Katalogi:

- t – sticky bit – powoduje, zmianę działania uprawnienia `w` dla katalogu — tylko właściciel katalogu lub konkretnego pliku może go usunąć (katalog /tmp), wartość numeryczna 01000
- sgid – nowo utworzone pliki będą mieć grupę właścicieli taką jak katalog. 02000

Pliki (tylko pliki wykonywalne – programy):

- sgid – niezależnie od tego, kto uruchamia program, proces będzie działał z uprawnieniami grupy właścicieli pliku programu. 02000
- suid – niezależnie od tego, kto uruchamia program, proces będzie działał z uprawnieniami właściciela pliku programu. 04000

Uprawnienia do plików VIII

umask

Wartość `umask` modyfikuje uprawnienia nowo tworzonych plików. Wartość ta mówi, których uprawnień nowe pliki mieć **nie** będą.
Jeśli program zażąda utworzenia pliku o uprawnieniach `perm`, to system utworzy plik o uprawnieniach `perm&(~umask)` (operatory bitowe jak w C).

ACL

POSIX definiuje listy kontroli dostępu (Access Control Lists), które pozwalają na ustawienie uprawnień bardziej szczegółowo niż dla właściciela i grupy właścicieli.
Można zdefiniować uprawnienia dla dowolnych grup i dowolnych użytkowników, uprawnienia domyślne dla nowych plików w katalogach.
Komendy: `getfacl`, `setfacl`