(Grundlagen der) Betriebssysteme | K.1

Franz J. Hauck | Institut für Verteilte Systeme, Univ. Ulm

K | Rechteverwaltung

(Grundlagen der) Betriebssysteme

Franz J. Hauck | Institut für Verteilte Systeme, Univ. Ulm

Überblick

Überblick der Themenabschnitte

- A – Organisatorisches
- B – Zahlendarstellung und Rechnerarithmetik
- C – Aufbau eines Rechnersystems
- D – Einführung in Betriebssysteme
- E – Prozessverwaltung und Nebenläufigkeit
- F – Dateiverwaltung
- G – Speicherverwaltung
- H – Ein-, Ausgabe und Geräteverwaltung
- I – Virtualisierung
- J – Verklemmungen
- K – Rechteverwaltung

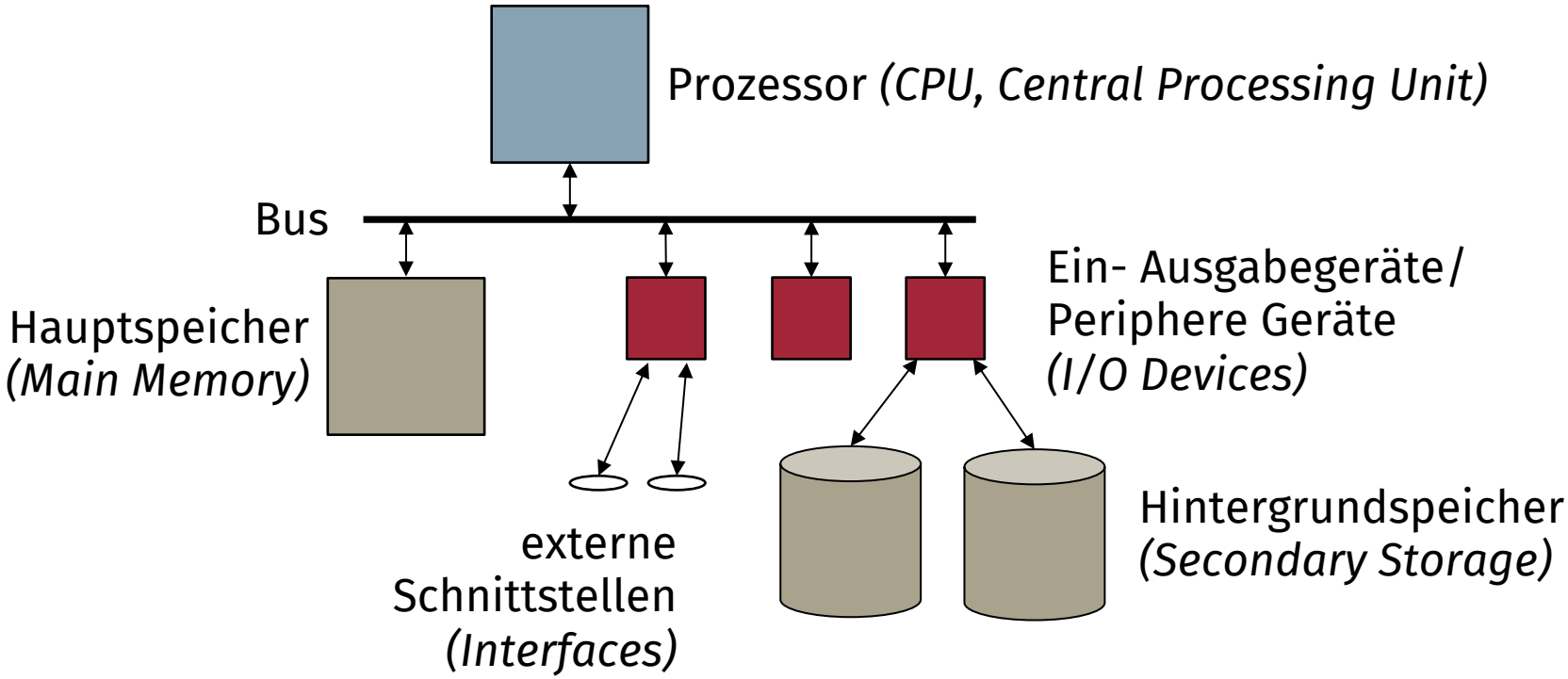
Inhaltsüberblick

Rechteverwaltung

- Berechtigungen
  - allgemeine Betrachtungen zur Rechte- und Nutzerverwaltung
- UNIX/Linux
  - Arten von Berechtigungen
  - Weitergabe von Privilegien
  - Anmeldung am System
  - erweiterte Berechtigungen
- Windows
  - ACLs unter NT- und POSIX-Systemen
- Schutz vor schädlicher/fehlerhafter Software

Einordnung

Alle physikalischen Ressourcen betroffen



Berechtigungen

Wer hat Zugriff oder darf bestimmte Dinge tun?

- Konzentration auf Betriebssysteme
  - Zugriff auf Betriebssystem-Ressourcen
  - Verwaltungsaufgabe des BS beinhaltet Rechteverwaltung
  - muss immer auf anderer Ebene ergänzt werden
    - z.B. Zugangsberechtigungen zu Räumen etc.
- Mehrprozesssysteme
  - nicht jeder Prozess soll alles können
- Mehrbenutzersysteme
  - nicht jeder Nutzer soll alles können

❖ Berechtigungssystem und dessen Durchsetzung erforderlich

Berechtigungen (2)

Zugriffsmöglichkeiten

- datenzentriert:
  - lesender und schreibender Zugriff
    - z.B. Datei lesen, Platte beschreiben, Speicher auslesen
- operationszentriert:
  - beliebige Operation
    - z.B. Datei löschen, Software installieren, Netzwerk nutzen, Prozess anhalten, Prozesse auflisten, Bildschirm nutzen, Drucken etc.
    - aber auch lesen und schreiben

Berechtigungen (3)

„Meta“-Berechtigungen

- Rechte vergeben, ändern und entziehen
  - pauschal
  - nur für bestimmte Rechte

Rechteverwaltung

Akteure einer Rechteverwaltung

- Subjekte
  - Wer kann etwas tun?
  - z.B. Benutzer
- Objekte
  - Was kann bearbeitet werden?
  - z.B. Datei
- Operationen
  - Was wird getan?
  - z.B. lesender Zugriff

Rechteverwaltung (2)

Organisation der Berechtigungen

- Rechte pro Subjekt
  - An welchen Objekten können welche Operationen ausgeführt werden?
  - z.B. pro Benutzer gespeichert, was erlaubt ist
  - Capability
    - eine Art Gutschein, der bestimmte Operation auf bestimmtem Objekt erlaubt

Rechteverwaltung (3)

Organisation der Berechtigungen (fortges.)

- Rechte pro Objekt
  - Welche Subjekte dürfen hier welche Operationen ausführen?
  - z.B. an Datei gespeichert, was erlaubt ist
  - Access Control List (ACL)
    - Liste der Subjekte und deren ausführbaren Operationen

Rechteverwaltung (4)

Organisation der Berechtigungen (fortges.)

- Rechte pro Operation
  - Welche Subjekte dürfen diese Operation an welchen Objekten ausführen?
  - z.B. an Löschoperation sind Nutzer und deren löschbare Dateien angehängt
  - sehr selten implementiert

Rechteverwaltung (5)

Positive Rechte

- Subjekt darf nichts
  - Rechte werden mit Capabilities oder ACLs hinzugefügt

Negative Rechte

- Subjekt darf alles
  - Capabilities und ACLs schränken ein

Kombination

- Subjekt erhält Rechte und wird wieder eingeschränkt
  - Reihenfolge von positiven und negativen Rechten entscheidend

Benutzerverwaltung

Unterscheidung von Benutzern

- Identifikation als Subjekte im System
  - zur Durchsetzung von Berechtigungen
- Hinterlegen individueller Merkmale
  - z.B. Name, Passwort, Heimatverzeichnis, organisatorische Zuordnung, Präferenzen etc.

Problem

- viele Rechte sind für Gruppen von Benutzern identisch
  - z.B. Studierende vs. Mitarbeiter, Mitglieder einer Abteilung
  - einzelne Vergabe der Rechte aufwändig

Benutzerverwaltung (2)

Unterscheidung von Benutzergruppen

- Einführung von Gruppen als weitere Subjekte
  - Berechtigungen an Gruppen festgemacht
    - erheblich vereinfacht
      - Rechteänderung wirkt auf alle Nutzer in der Gruppe
  - Benutzerzuordnung zu Gruppen
    - Zugehörigkeit zu Gruppen aus dem realen Leben
      - z.B. Studierende, Professoren, Administratoren, Sopra-Gruppe
    - Gruppierung dediziert zur Rechtevergabe
      - z.B. Gruppe, die USB-Stick verwenden darf, Gruppe, die drucken darf
  - bedarf Gruppenverwaltung
    - häufig Zuordnung zu mehreren Gruppen möglich

Benutzerverwaltung (3)

Erkennung des Benutzers durch das System

- Identifikation über Benutzername
  - Beweis der Identität durch weiteres Kriterium
    - meist geheimes Passwort
    - Alternativen zum Passwort: Chipkarte, Iris-Scan, Fingerabdruckscan, ...
- verschiedene Verfahren
  - einen Gegenstand haben
  - ein biometrisches Merkmal haben
  - etwas wissen
- Kombinationen mehrerer Verfahren sicherer
- Benutzername + Authentifizierungskriterium = Credentials

Benutzerverwaltung (4)

Anmeldedaten

- Speicherung in sicherer Datenbank
  - vor Manipulation und illegalem Auslesen geschützt
  - aber Änderungen müssen möglich sein
    - z.B. regelmäßige Passwortänderungen
- Abgleich durch den Anmeldeprozess bei der Anmeldung
  - sichere Datenbank muss für Anmeldeprozess lesbar sein

Änderungen an Benutzer-/Gruppendatenbanken

- benötigt erweiterte Rechte
  - z.B. Windows: GUI-Tool zur Kontoverwaltung
  - z.B. Kubuntu: KDE-Kontrollmodul „User-Management“
  - z.B. Linux: `passwd`-Kommando

(Grundlagen der) Betriebssysteme | K.2

Franz J. Hauck | Institut für Verteilte Systeme, Univ. Ulm

Inhaltsüberblick

Rechteverwaltung

- Berechtigungen
  - allgemeine Betrachtungen zur Rechte- und Nutzerverwaltung
- UNIX/Linux
  - Arten von Berechtigungen
  - Weitergabe von Privilegien
  - Anmeldung am System
  - erweiterte Berechtigungen
- Windows
  - ACLs unter NT- und POSIX-Systemen
- Schutz vor schädlicher/fehlerhafter Software

UNIX/Linux Rechteverwaltung

Berechtigungen geknüpft an Inode (Objekt)

- alles, was einen Inode hat, kann im Zugriff eingeschränkt werden
  - Dateien, Verzeichnisse, Geräte u.a.

Benutzer (Subjekt)

- Nutzer durch eindeutige Nummer (User-ID, UID) repräsentiert
  - Nutzer 0 (root) darf alles
    - Rechteprüfung ausgeschaltet
    - Administrator

Nutzergruppen (Subjekt)

- Gruppen durch eindeutige Nummer (Group-ID, GID) repräsentiert

UNIX/Linux Rechteverwaltung (2)

Inodes und Berechtigungen

- Inodes haben einen Eigentümer (UID)
- Inodes gehören genau einer Gruppe an (GID)
- Inodes speichern Berechtigungen für drei Kategorien:
  - Eigentümer *(User)*
  - Gruppe *(Group)*
  - alle anderen (Rest der Welt, *Others*)

  - es gilt die erste Kategorie, die zutrifft
  - Berechtigungen nur vom Eigentümer änderbar
    - oder von root

UNIX/Linux Rechteverwaltung (3)

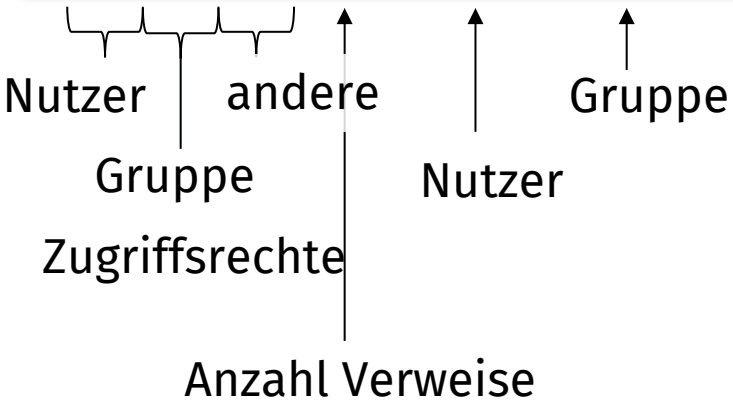
Berechtigungen

- jeweils für die drei Kategorien individuell vergebbar
- Datei
  - Leserecht (r), Schreibrecht (w), Ausführungsrecht (x)
- Verzeichnis
  - Leserecht (r), Schreibrecht (w), Durchgangsrecht (x)
  - Leserecht erlaubt Auflisten des Verzeichnisinhalts
  - Schreibrecht erlaubt Löschen und Anlegen von Verweisen
  - Durchgangsrecht erlaubt Auflösen von Pfadnamen zu Inodes
    - d.h. Name im Verzeichnis muss bekannt sein

UNIX/Linux Rechteverwaltung (4)

Beispiel

```
$ ls -l /home/alice
total 40
drwx------ 3 alice users  4096 Jun 14 14:05 Musik
-r--r--r-- 1 alice users   120 Jun 14 12:11 bsp.text
-rwxr-x--- 1 alice users 14828 Jun 11 00:42 test
-rw-r----- 1 alice users  2035 Jun 11 00:41 test.c
```



UNIX/Linux Rechteverwaltung (5)

Weiteres Beispiel

```
$ ls -ld /tmp/test
drw------- 2 alice users 4096 Jun 14 07:11 /tmp/test
```

- Nutzerin Alice hat Leserecht im Verzeichnis
  - Auflistung ergibt:

```
$ ls -l /tmp/test
ls: cannot access test/welt: Permission denied
ls: cannot access test/hallo: Permission denied
total 0
-????????? ? ? ? ?            ? hallo
-????????? ? ? ? ?            ? welt
```

  - Pfade können nicht aufgelöst werden (kein Durchgangsrecht)
    - daher kein Zugang zum Inode und seinen Metadaten

UNIX/Linux Rechteverwaltung (6)

Weitere Berechtigungen an Inodes

- User S-Bit
  - bei ausführbaren Dateien
    - Ausführung findet unter Berechtigung des Dateieigentümers statt, nicht unter der Berechtigung des Nutzers
- Group S-Bit
  - bei ausführbaren Dateien
    - Ausführung findet unter Gruppenberechtigung der Datei statt nicht unter den Gruppen des Nutzers
  - bei Verzeichnissen
    - neue Einträge bekommen automatisch selbe Gruppenzuordnung wie Verzeichnis

UNIX/Linux Rechteverwaltung (7)

Weitere Berechtigungen an Inodes (fortges.)

- Sticky Bit (T)
  - bei ausführbaren Dateien
    - Speicherseiten werden nicht ausgelagert
    - heute nicht mehr unterstützt
  - bei Verzeichnissen
    - trotz Schreibrecht kann ein Nutzer nur seine eigenen Einträge löschen
    - wird z.B. für Verzeichnis temporärer Dateien genutzt

UNIX/Linux Rechteverwaltung (8)

Darstellung zusätzlicher Rechte

```
$ ls -l /home/alice/demo
total 48
-rws--x--- 1 alice users  4096 Jun 14 14:05 cmd1
-rwxr-sr-x 1 alice sopra  2088 Jun 14 12:11 cmd2
drwxrws--- 3 alice sopra   503 Jun 14 12:11 shared
drwxrwxrwt 1 alice users 14828 Jun 11 00:42 stickydir
-rwSr-S--T 1 alice users  2035 Jun 11 00:41 zombie
```

- S-Bits nur sinnvoll bei gleichzeitigem X-Bit
- Großschreibung falls nicht gesetzt

UNIX/Linux Rechteverwaltung (9)

Beispiel Gruppenarbeit

```
$ ls -la /home/alice/demo/shared
total 24
drwxrws---  3 admin sopra  4096 Feb 21 00:54 .
drwxr-xr-x 56 root  root  86016 May 26 23:21 ..
-rwxrwx---  2 alice sopra  4096 Jun 14 11:09 fromAlice
-rw-rw----  1 bob   sopra 24476 Jun 14 10:48 fromBob
-rwxrwx---  1 emil  sopra 96061 Jun 10 20:14 fromEmil
```

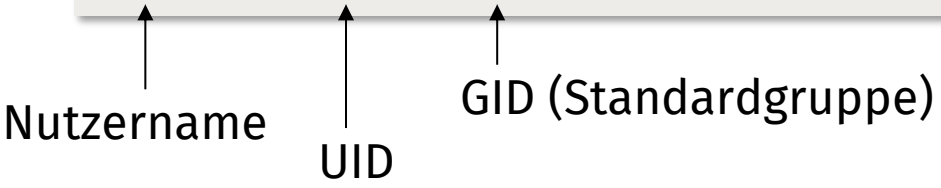
- alle neuen Dateien bekommen `sopra`-Gruppe
  - obwohl Dateien sonst mit Standardgruppe des jeweiligen Nutzers angelegt werden (hier `users`)

UNIX/Linux Benutzerverwaltung

Speicherung der Passwörter

- /etc/passwd – öffentlich lesbar
  - enthält nur nicht-sensitive Informationen
    - Benutzername, Ids, Heimatverzeichnis, genutzte Shell

```
root:x:0:0:root:/root:/bin/sh
nico:x:1000:1000:Nico:/home/nico:/bin/bash
```



- /etc/shadow – nicht öffentlich lesbar
  - enthält Passwort

```
root:$6$Az5Ne3iKJ.2B/…:15475:0:::::
nico:$6$Kjsd638/.JehD…:15458:0:90:5:::
```

UNIX/Linux Benutzerverwaltung (2)

Speicherung der Gruppen

- `/etc/group` – öffentlich lesbar
  - enthält nur nicht-sensitive Informationen
    - Gruppenname, ID, zugeordnete Nutzer (außer Standardgruppe)
- `/etc/gshadow` – nicht öffentlich lesbar
  - enthält optionales Passwort
  - Gruppen können mit Passwort geschützt sein
    - Anmelden an der Gruppe macht Berechtigung verfügbar

Übertragung von Privilegien in UNIX/Linux

Prozesstabelle

- Betriebssystem führt Prozesstabelle aktiver Prozesse
- Prozess hat eine Benutzer- und Gruppen-ID (UID und GID)
  - operiert mit den Rechten des Benutzers und der Gruppe

ID-Konzept unter Unix

- reale UID, reale GID
  - IDs, unter denen der Prozess gestartet wurde
- effektive UID, effektive GID
  - IDs, gegen die geprüft wird, ob Zugriffe erlaubt sind
- normalerweise reale und effektive IDs identisch
  - Prozess erbt alle vier vom Elternprozess

Übertragung von Privilegien in UNIX/Linux (2)

Änderungen der IDs

- Systemaufrufe `setuid()` / `setgid()`
  - setzen reale und effektive ID des Prozesses
  - nur `root` kann auch reale ID setzen
  - normaler Nutzer kann nur seine eigene UID/GID setzen oder die mit S-Bit in der ausgeführten Datei gesetzten IDs
    - d.h. Umschaltung zwischen S-Bit und eigener UID möglich

Übertragung von Privilegien in UNIX/Linux (3)

Beispiel: Passwort ändern

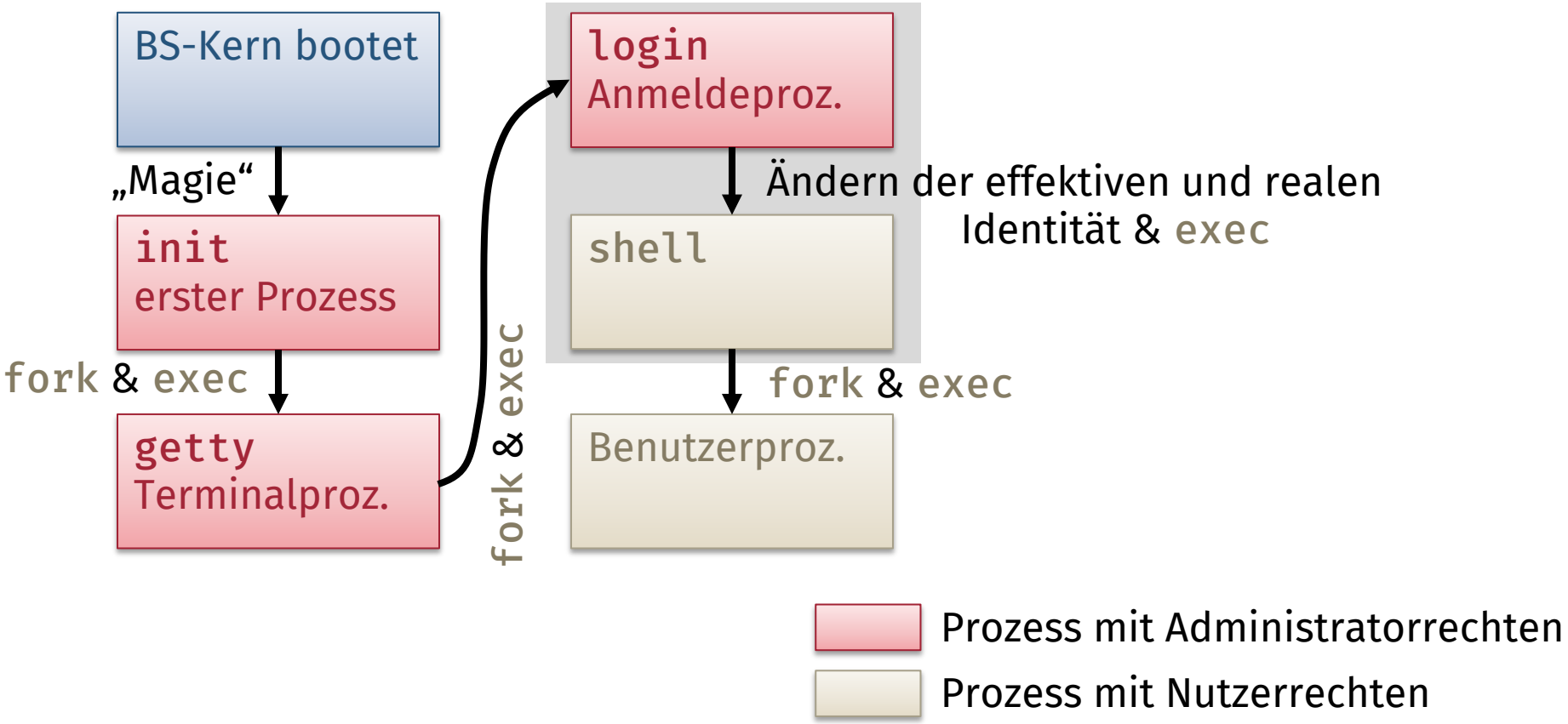
- jeder Benutzer soll sein Passwort ändern können
  - Schreibzugriff auf die Passwort-Datei `/etc/shadow` erforderlich
  - aber nicht jeder soll Passwörter lesen und die anderer Nutzer schreiben können
- ◆ Lösung
  - Programm zur Passwortänderung hat Administrator-S-Bit

```
$ ls -l /usr/bin/passwd
-rwsr-xr-x  1 root root 42824 Apr  9 04:32 /usr/bin/passwd
```

  - wird mit Administratorrechten ausgeführt,
    - effektive UID = 0

Anmeldung unter UNIX/Linux

Ablauf des Login-Prozesses



Anmeldung unter UNIX/Linux (2)

Login-Prozess

- läuft mit Administratorrecht
  - kann Passwortdatei lesen
  - fragt Passwort ab, verschlüsselt es und vergleicht die beiden
  - startet Shell mit effektiver und realer UID/GID des angemeldeten Nutzers

```
/* cryptedpwd, uid, gid, shell aus /etc/{passwd,shadow} */
password = getpass("Password:");

if (strcmp(crypt(password, cryptedpwd), cryptedpwd) != 0) {
    write(1, "Login incorrect\n", 16);
} else {
    /* richtiges Passwort, starte Shell für Benutzer */
    if (fork()==0) {
        setuid(uid); setgid(gid); exec(shell);
    }
}
```

(Grundlagen der) Betriebssysteme | K.3

Franz J. Hauck | Institut für Verteilte Systeme, Univ. Ulm

Inhaltsüberblick

Rechteverwaltung

- Berechtigungen
  - allgemeine Betrachtungen zur Rechte- und Nutzerverwaltung
- UNIX/Linux
  - Arten von Berechtigungen
  - Weitergabe von Privilegien
  - Anmeldung am System
  - erweiterte Berechtigungen
- Windows
  - ACLs unter NT- und POSIX-Systemen
- Schutz vor schädlicher/fehlerhafter Software

Erweiterte Berechtigungen in Linux

Lange Access Control Lists im Dateisystem

- Implementierung der POSIX-ACLs
  - Speicherung in sog. Extended Attributes (xattr; Name-Wert-Paare)
  - im Dateisystem für jeden Inode
- Markierung im Verzeichnislisting
  - klassische Rechte, dahinter „+“

```
$ ls -l /srv/www/web1
-rwxrwx---+ 2 root root  4096 May 18 14:16 /srv/www/web1
```

- positive Rechte für Benutzer und Gruppen
- Eigentümer kann Rechte selbst setzten
  - für beliebige Nutzer und Gruppen

Erweiterte Berechtigungen in Linux (2)

Bearbeiten der langen ACLs

- Setzen/Löschen über den Befehl `setfacl`

```
$ setfacl –m user:alice:rwx /srv/www/web1
$ setfacl –m group:webusers:rwx /srv/www/web1
```

- Anzeige der ACLs über den Befehl `getfacl`

```
$ getfacl /srv/www/web1
user::rwx
user:alice:rwx
group::---
group:webusers:rwx
mask::rwx
other::---
```

Erweiterte Berechtigungen in Linux (3)

Besonderheiten

- ACL-Maske (maximale Rechte)
  - z.B. `mask::rwx`
  - bitweises logisches Und auf alle ACL-Einträge vor der Zugriffsprüfung
  - erlaubt zentrale Einschränkung der Rechte
  - Maske wird an Stelle der Gruppenrechte angezeigt

Erweiterte Berechtigungen in Linux (4)

Besonderheiten (fortges.)

- Vererbung der ACLs
  - Default-ACLs
    - z.B. `default:user:alice:rwx`
    - z.B. `default:group::---`
  - wird neue Datei angelegt, werden Default-ACLs des enthaltenden Verzeichnisses übernommen
  - wird neues Verzeichnis angelegt, werden die Default-ACLs des enthaltenden Verzeichnisses als Default-ACLs des neuen gesetzt
  - nachträgliche Änderungen der ACLs immer möglich

Inhaltsüberblick

Rechteverwaltung

- Berechtigungen
  - allgemeine Betrachtungen zur Rechte- und Nutzerverwaltung
- UNIX/Linux
  - Arten von Berechtigungen
  - Weitergabe von Privilegien
  - Anmeldung am System
  - erweiterte Berechtigungen
- Windows
  - ACLs unter NT- und POSIX-Systemen
- Schutz vor schädlicher/fehlerhafter Software

Berechtigungen unter Windows

Berechtigungen hängen an Objekten

- alles ist Objekt, insbesondere Dateien und Verzeichnisse
- Security Descriptors für alle Objekte
  - ACLs an Security Descriptors gekoppelt
- NTFS kann Security Descriptors speichern
- FAT kann das nicht
  - erlaubt lediglich Schreibschutz

Berechtigungen unter Windows (2)

Rechtestufen

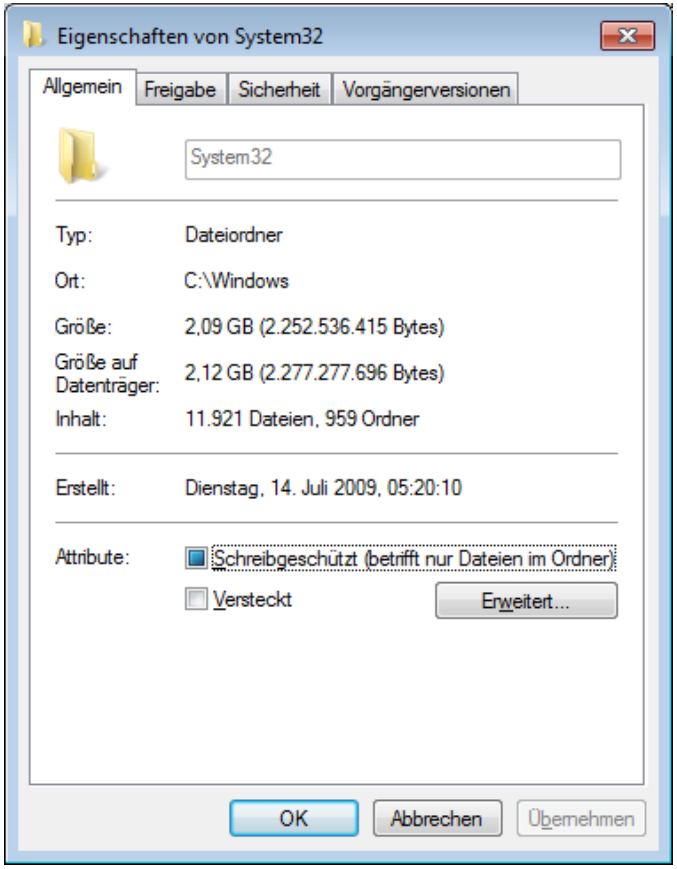
- No access: Kein Zugriff
- List: Anzeige von Dateien in Verzeichnissen
- Read: Inhalt von Dateien lesen, und list
- Add: Hinzufügen von Dateien zu einem Verzeichnis, und list
- Read & Add: Wie read und add
- Change: Ändern von Dateiinhalten, Löschen von Dateien und read & add
- Full: Ändern von Eigentümer und Zugriffsrechten und change

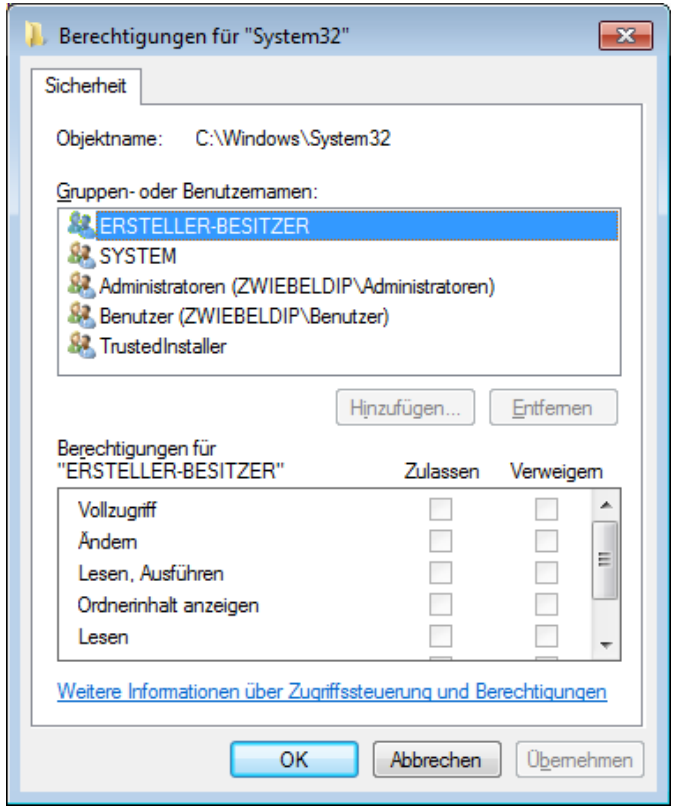
ACLs konfigurierbar für beliebige Benutzer und Gruppen

- positive und negative ACLs

Berechtigungen unter Windows (3)

Einstellung der Rechte





Berechtigungen unter Windows (4)

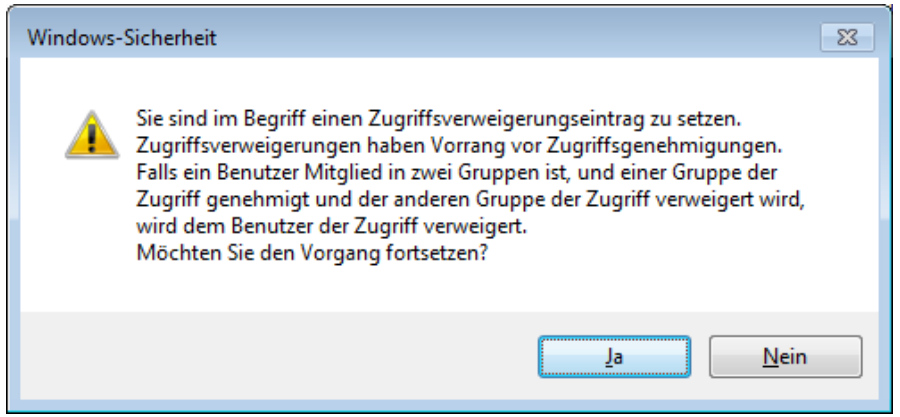
Weitere Anmerkungen

- Benutzer SYSTEM (= Betriebssystem) kann explizit Rechte zugewiesen bekommen
  - normalerweise Vollzugriff auf alles
- Administratoren dürfen in der Regel ebenfalls nicht alles
  - müssen auch ausdrücklich Rechte zugewiesen bekommen
- weit komplexeres und detaillierteres Rechtekonzept als unter Linux
- Vererbung der ACLs an niedrigere Ebenen im Verzeichnisbaum ist Standard, lässt sich abschalten

Berechtigungen unter Windows (5)

Auswertung der ACLs

- ist keine ACL vorhanden, ist der Zugriff nicht möglich
- Verbote überschreiben Erlaubnisse
  - negative Rechte überschreiben positive Rechte
  - Achtung bei Verboten für ganze Gruppen
  - z.B. Alice ist in der Gruppe der Studierenden, Bob nicht
    - Alice legt Datei an
    - sie setzt ACLs:
      Alice & Bob dürfen lesen,
      Studierenden ist alles verboten
    - Resultat: Alice darf selbst nicht mehr lesen!



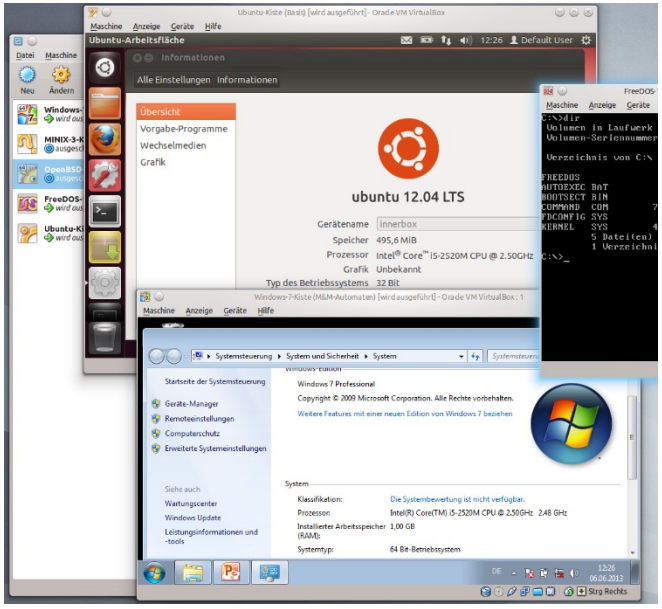
Schutz vor schädlicher/fehlerhafter Software

Speicherbereiche schützen

- logische Adressräume (vgl. Kapitel G)

Weitere Maßnahmen zur Isolation von Prozessen

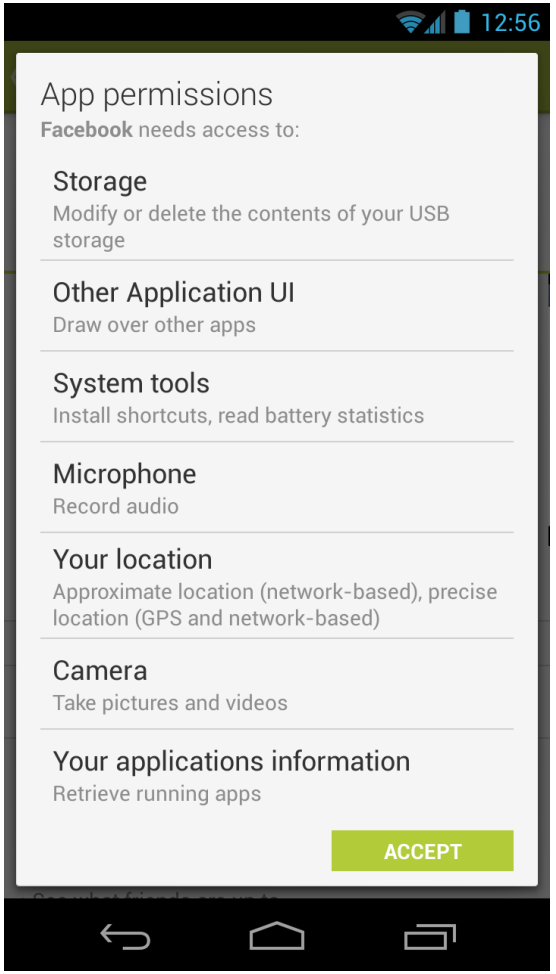
- Application Sandboxing
  - z.B. bei Android: Prozesse voneinander abgeschottet
- `chroot`
  - Trennung auf Dateisystemebene
  - neues Wurzelverzeichnis
- Virtualisierung
  - siehe nächstes Kapitel J



Schutz vor schädlicher/fehlerhafter Software (2)

Beispiel: Android Sandbox

- Sandbox basiert auf klassischen UNIX-Rechten
- jeder App läuft als Prozess mit eigener UID
  - keine zwei Prozesse mit gleicher UID
  - Prozess kann standardmäßig nur auf eigene Ressourcen zugreifen
  - Prozess kann nach erweiterten Rechten fragen, um auf andere Ressourcen/Daten auch anderer Prozesse zuzugreifen



Vertiefung/Weiterführung der Thematik

Uni-Veranstaltungen

- Grundlagen der Rechnernetze / Vernetzte Systeme
- Sicherheit in IT-Systemen
- Web Engineering
- Grundlagen Verteilter Systeme / Verteilte Systeme
- Kryptologie

Literatur

- Barrett, Silverman, Byrnes. *Linux Security Cookbook*. O'Reilly, 2003.