

Studiengang Kommunikationsinformatik
Studiengang Praktische Informatik
Prof. Dr.-Ing. Damian Weber
Dipl.-Inf. Marion Bohr
Daniel Henry, M.Sc.
Dipl.-Ing. Michael Sauer

Systemmanagement und Sicherheit

6. Übung

Aufgabe 1 (Signal-Handling `signal()`)

Schreiben Sie ein C-Programm `sigtest`, das einen einzigen Signalhandler für alle möglichen Signale besitzt. Der Signalhandler soll mittels `signal(3)` aktiviert werden.

Mittels `sleep(3)` soll `sigtest` für eine Minute existieren und innerhalb von `main` folgenden Return-Code zurückgeben:

- 0, falls kein Signal innerhalb dieser Minute empfangen, oder
- die Signalnummer

Die Kommunikation zwischen Signal-Handler und `main()`, um die Signalnummer mitzuteilen, soll über eine globale *volatile int* Variable `signo` realisiert werden.

Aufgabe 2 (Signal-Handling `sigaction()`)

Schreiben Sie analog zur vorherigen Aufgabe ein Programm `sigtest2`, das den gleichen Mechanismus mit Hilfe von `sigaction(2)` implementiert.

Aufgabe 3 (Signal-Handling (Signale künstlich erzeugen))

Benutzen Sie Ihr Programm `sigtest2` als Grundlage für ein Programm `sigtest3`, das mit einem erhöhten Parameter für `sleep()` arbeitet, um vier der in der Vorlesung angegebenen Signale zu simulieren. Hierbei bedeutet *Simulieren* nicht, dass mit Hilfe von `kill` oder `kill()` die jeweilige Signalnummer erzeugt wird. Vielmehr sollen beim Ablauf des Programms `sigtest` die Bedingungen erzeugt werden, die zum Senden des Signals führen, etwa dass ein Alarmtimer abläuft (siehe `alarm(3)`).

In dem vom Internet abgeschirmten Testnetz 172.16.0.0 sind virtuelle Maschinen `play01.local` bis `play99.local` eingerichtet, die vom Rechner `isl-s-01` aus erreichbar sind. Jedem Benutzer ist eine Maschine `playnn` zugeordnet. Siehe auch die Dateien `/etc/hosts` und `/usr/sfw/cuda/etc/vm.conf` auf `isl-s-01`. Das `root`-Passwort der virtuellen Maschine erhalten Sie in Ihrer Übungsgruppe. Loggen Sie sich als `root` auf dem Ihnen zugeordneten `play`-Rechner ein. Auf den `play`-Rechnern gibt es die Editoren `ee` und `vi`, wobei `ee` der einfacher zu bedienende Editor sein könnte (die dort eingblendete Hilfe verwendet das `^`-Symbol für die `Strg`-Taste). Das Starten der virtuellen Maschine erfolgt, sofern Ihr Benutzername in `vm.conf` richtig eingetragen ist, von `isl-s-01` aus mittels `playStart -s`

Aufgabe 4 (User-Accounts (1))

Legen Sie sich auf dem `play`-Rechner einen eigenen persönlichen Login-Account an. Benutzen Sie hierfür Ihren Benutzernamen aus dem STL-Labor. Detaillierte Informationen zum Verwalten von Benutzern sind u.a. im FreeBSD-Handbook Kapitel *Users and Basic Account Management* enthalten.

Setzen Sie für die folgende Aufgabe für `joe` Ihren Benutzernamen ein. Jedes Teammitglied soll dies für seinen Loginnamen durchführen.

Hier nun *beispielhaft* die Vorgehensweise für den Benutzer `joe`

- legen Sie eine neue Gruppe `joe` an,
- legen Sie einen neuen Benutzer `joe` an, dieser soll als Hauptgruppe die Gruppe `joe` haben; hierbei ist `vipw` zu benutzen
- geben Sie `joe` ein Homeverzeichnis und ein Paßwort
- testen Sie den neuen Account mit Hilfe des `su` Kommandos und des `id` Kommandos
- loggen Sie sich als `joe` von `isl-s-01` aus auf dem `play`-Rechner ein
- schreiben Sie als `joe` ein C-Programm, das die numerische User-ID und den Inhalt der Umgebungsvariable `PATH` ausgibt.

Legen Sie einen weiteren Benutzer `joex` mit Hilfe des `adduser` Kommandos an.

Legen Sie einen weiteren Benutzer `joej` mit Hilfe des `pw` Kommandos an.

Aufgabe 5 (User-Accounts (2))

Legen Sie mit Hilfe des `adduser`-Kommandos (Option `-f`) und einem von Ihnen geschriebenen Skript 50 Benutzer an, die (in der Bedeutung von `joe` von Aufgabe 1) nun

- `joe01,...,joe50` heißen,
- Paßwörter `passjoe01,...,passjoe50` haben,
- Homeverzeichnisse `/home/joe01,...,/home/joe50` besitzen
- als Login-Shell `/bin/tcsh` benutzen.

Bei dieser Aufgabe ist das `jot` Programm hilfreich. Ungeduldige können

```
jot -w %02d 20 1 20
```

ausprobieren, bevor sie die Manualpage lesen.

Aufgabe 6 (User–Accounts (3))

Erweitern Sie Ihr Skript aus Aufgabe 2, damit es sicherere Passwörter generiert. Wenn Sie sicherere Paßwörter generieren möchten, nutzen Sie

```
openssl rand -base64 6
```

Aufgabe 7 (at–Kommando)

Benutzen Sie als User joe01 das at–Kommando, um einen Befehl in der Zukunft ausführen zu lassen. Mit dem mailx Kommando können Sie die e–Mail lesen, die die Standardausgabe des ausgeführten Kommandos enthält.

Aufgabe 8 (crontab–Kommando)

Benutzen Sie als User joe01 das crontab–Kommando, um einen Befehl periodisch ausführen zu lassen.