

BAI3 BSP	Praktikum Betriebssysteme	Hübner/Nogalski
SS 2021	Aufgabe 1 – UNIX (Linux)	Seite 1 von 2

1. bash – Shell (Linux-Befehlsinterpreter)

Machen Sie sich (wieder) mit der **bash-Shell** unter Linux vertraut.

Informationen zur bash und UNIX/Linux-Befehlen finden Sie in den beigefügten Befehlsübersichten, auf einem Linux-Rechner (durch die Befehle „**apropos**“ und „**man**“), in Büchern und im Internet, z.B. unter

- <https://wiki.ubuntuusers.de/Shell/Befehls%C3%BCbersicht/>
- <http://www.shellbefehle.de>
- <http://linuxwiki.de/LinuxKommandos>
- https://en.wikipedia.org/wiki/List_of_Unix_commands

Unten finden Sie außerdem Tipps, wie Sie kostenlos auf ein Linux-System zugreifen können.

2. Shell-Skripte

Die bash-Shell ist in der Lage, mehrere Befehle aus einer Textdatei („Shell-Skript“) zu lesen und auszuführen. In der Dokumentation finden Sie dazu auch die Beschreibung einfacher Kontrollstrukturen (if, for, ...) sowie von Funktionsaufrufen (z.B. <http://www.selflinux.org/selflinux/html/shellprogrammierung.html>). Ein Beispiel-Skript `example.sh` finden Sie im Pub-Verzeichnis.

Erstellen Sie folgende Shell-Skripte (bash) und testen Sie diese in einem Linux-Terminal. Für das Protokoll reicht der kommentierte Source Code des Skripts. Während der Abnahme sollten Sie den Code erklären können.

a)

`frename directory string`

Das Skript **frename** hängt für alle Dateien und Verzeichnisse im angegebenen Unterverzeichnis directory die Zeichenkette string an den aktuellen Dateinamen (Umbenennung). Sie können davon ausgehen, dass in den Dateinamen keine Leerzeichen vorkommen. Die Umbenennung soll sich nur auf das angegebene Verzeichnis beziehen (keine Berücksichtigung von weiteren Unterverzeichnissen).

Beispiel (im Unterverzeichnis `mydir` soll an alle Dateinamen „-v02“ angehängt werden):

```
$ frename mydir -v02
```

b)

`try-host.sh hostname`

Der in der Befehlszeile angegebene Rechner (Hostname) soll auf Erreichbarkeit hin überwacht werden. Dazu sendet das Skript einmal pro Sekunde ein "ping" an den angegebenen Rechner (nur ein ping Paket) und wertet den return value (auch „exit code“) aus (siehe `man ping`). War der ping Befehl erfolgreich, wird der Hostname mit einem OK- Vermerk ausgegeben, andernfalls wird er mit einem FAILED-Vermerk ausgegeben.

Beispiel: Der Aufruf

```
bash try-host.sh google.de
```

erzeugt jeweils nach einer Sekunde Wartezeit eine Ausgabe der Art:

```
google.de OK
```

falls ein Rechner unter dem Hostnamen `google.de` erreichbar ist, anderenfalls

```
google.de FAILED
```

Tipp: Sie können zum Testen als Hostnamen „localhost“ (d.h. den eigenen Rechner) verwenden, um die Erreichbarkeit zu garantieren.

BAI3 BSP	Praktikum Betriebssysteme	Hübner/Nogalski
SS 2021	Aufgabe 1 – UNIX (Linux)	Seite 2 von 2

3. C-Programm mit Systemaufrufen

Laden Sie das Programm `hello.c` aus dem Pub-Verzeichnis herunter. Es wird mit dem Befehl `gcc -o hello hello.c` übersetzt, so dass anschließend das ausführbare Programm `hello` vorhanden ist.

Das zu erstellende C-Programm **saveuser** soll folgendes leisten:

1. Eine Eingabeaufforderung anzeigen
2. Einen Dateinamen (maximal 30 Zeichen) von der Tastatur lesen
3. Eine Datei mit diesem Namen erzeugen
4. Die Userid des aktuellen Benutzers mittels Systemaufruf (System Call) ermitteln und diese Informationen in die Datei schreiben
5. Eine Erfolgsmeldung oder eine Fehlermeldung (falls das Erzeugen der Datei nicht funktioniert hat) auf dem Bildschirm ausgeben.

Beispiel (Ausgaben sind *kursiv* dargestellt):

```
$ saveuser
```

Name der neuen Datei: *bsp1*

Die Datei *bsp1* wurde erfolgreich erzeugt!

Der Inhalt der Datei sollte anschließend z.B. so aussehen (Userid):

1034

Benutzen Sie zur Realisierung der Systemaufrufe folgende C-Bibliotheksfunktionen (Dokumentation auch über „man“-Befehl erhältlich):

- `fgets` : Liest eine Zeichenkette von `stdin` ein (*Achtung: inkl. Newline!*)
- `fopen` : Legt eine Datei an und öffnet sie
- `fprintf` : Schreibt einen String in eine Datei
- `fclose` : Schließt eine Datei

Hinweise (zur Erinnerung ;-):

- Ggf. muss die C-Bibliotheksfunktion explizit in Ihren Programmcode eingebunden werden (`#include`)
- Ein String wird in C als `char`-Array (z.B. `char myarray[40]`) repräsentiert. Der Name des Arrays (hier: `myarray`) liefert in C die Adresse des ersten Elements, ist also ein Pointer auf einen Speicherplatz vom Typ `char` (der Typ von `myarray` ist also `char*`).
- Die Funktion `int strlen(char* string)` liefert als Rückgabewert die Länge eines Strings.
- Ein String wird in C durch den ASCII-Wert 0 nach dem letzten Zeichen abgeschlossen (erhöht den Speicherplatzbedarf des `char`-Arrays um ein Byte!). Daher wird z.B. durch eine Zuweisung `myarray[5] = 0` der String `myarray` für alle String-Verarbeitungsfunktionen auf die ersten fünf Zeichen verkürzt (0 – 4), obwohl physikalisch mehr Speicherplatz reserviert wurde.

Frage: Wieviele Systemaufrufe („System Calls“) werden durch das Programm `saveuser` bei einer Ausführung mindestens erzeugt?