BS – aUFGABE 1

**WAS IST EINE TAB-EXPANSION**? Eine Tab-Expansion ist eine Autovervollständigung. Dabei erweitert die Funktion die aktuelle Eingabe eines Dateipfads zum Beispiel.

**Was erhalten sie bei beim Drücken der Tastenkombination <Alt><.>?:** Tastenkombination <Alt><.> liefert aus dem letzten aufgerufenen Shell-Befehl die letzten Zeichenketten.

**Geben Sie das Verzeichnis nach Erweiterung sortiert aus:**

linux-8b19:~ # ls -x

.Xauthority .adobe .bash\_history

.cache .config .dbus

.dmrc .esd\_auth .gconf

.gnome2 .gnupg .gstreamer-0.10

.gtkrc-2.0-kde4 .kbd .kde4

.local .macromedia .mozilla

.rcc .skel .vboxclient-clipboard.pid

.vboxclient-display.pid .vboxclient-draganddrop.pid .vboxclient-seamless.pid

.xsession-errors Arbeitsfläche Bilder

Bs\_Prakt Dokumente Downloads

Musik Videos Vorlagen

bin hello hello.c

inst-sys splitfix.sh test

~Bs\_Prakt Öffentlich

**Geben Sie das Verzeichnis nach Modifikationszeit sortiert aus**:

linux-8b19:~ # ls -t

.xsession-errors .adobe .gstreamer-0.10 Videos

konsole.txt .bash\_history .rcc Downloads

.gconf .mozilla .gnome2 Vorlagen

.vboxclient-draganddrop.pid .cache .esd\_auth Öffentlich

.vboxclient-seamless.pid .config .kde4 .dmrc

.vboxclient-display.pid hello .local .dbus

.vboxclient-clipboard.pid hello.c .skel .kbd

.Xauthority Dokumente .gtkrc-2.0-kde4 inst-sys

test Arbeitsfläche .gnupg bin

splitfix.sh ~Bs\_Prakt Bilder [...]

**Kehren sie für beide sortiervarianten die Rheinfolge um:**

linux-8b19:~ # ls -r

Öffentlich Musik .skel .gconf

~Bs\_Prakt Downloads .rcc .esd\_auth

test Dokumente .mozilla .dmrc

splitfix.sh Bs\_Prakt .macromedia .dbus

konsole.txt Bilder .local .config

inst-sys Arbeitsfläche .kde4 .cache

hello.c .xsession-errors .kbd .bash\_history

hello .vboxclient-seamless.pid .gtkrc-2.0-kde4 .adobe

bin .vboxclient-draganddrop.pid .gstreamer-0.10 .Xauthority

Vorlagen .vboxclient-display.pid .gnupg

Videos .vboxclient-clipboard.pid .gnome2

**GEBEN SIE DAS VERZEICHNIS REKURSIV MIT UNTERVERZEICHNISEN AUS:** Der Befehl lautet: ls –R

**Was passiert beim Befehl ls –l | sort –rnk5?:** Der Aufruf sortiert die Dateiliste rückwärts und numerisch nach der 5. Spalte

r- Steht für reverse, umdrehen der Sortierreihenfolge.

n- Sortierung nummerisch/nummerischer Stringwert

k- Nach Key, Index der Spalten

# ls -l

total 44

-rw-r--r-- 1 root root 79 Oct 19 19:01 .~lock.Bs\_Prak\_1.odt#

-rw-r--r-- 1 root root 22977 Oct 19 19:01 Bs\_Prak\_1.odt

lrwxrwxrwx 1 root root 9 Oct 19 19:03 ltext3.txt -> text3.txt

-rw-r--r-- 1 root root 30 Oct 19 18:49 text04.txt

-rw-r--r-- 1 root root 30 Oct 19 18:47 text1.txt

-rw-r--r-- 1 root root 30 Oct 19 18:48 text2.txt

-rw-r--r-- 1 root root 31 Oct 19 19:03 text3.txt

**Was passiert, wenn sie ltext3.TXT VERÄNDERN BEI TEXT3.TXT? WIE VERHÄLT ES SICH?:** Die Ursprungsdatei wird ebenfalls geändert

# ls -l

total 44

-rw-r--r-- 1 root root 79 Oct 19 19:01 .~lock.Bs\_Prak\_1.odt#

-rw-r--r-- 1 root root 22977 Oct 19 19:01 Bs\_Prak\_1.odt

lrwxrwxrwx 1 root root 9 Oct 19 19:03 ltext3.txt -> text3.txt

-rw-r--r-- 1 root root 30 Oct 19 18:49 text04.txt

-rw-r--r-- 1 root root 30 Oct 19 18:47 text1.txt

-rw-r--r-- 1 root root 30 Oct 19 18:48 text2.txt

-rw-r--r-- 1 root root **31 Oct 19 19:03 text3.txt**

Es ist sichtbar, dass die Datei text.3.txt verändert wurde.

**Was passiert, wenn sie ltext02.txt löschen?:** Der symbolische Link wird nur gelöscht (nachzusehen mit ls –l), allerdings wird die Urspungsdatei nicht gelöscht.

**Was passiert, wenn sie text3.txt löschen?** Da wir die Urspungsdatei löschen, zeigt der symbolische Link auf eine entfernte Datei.

**Shell** **Sonderzeichen**

Demonstrieren Sie die Platzhalterzeichen mit eigenen Beispielen

linux-8b19:~/Bs\_Prak # **ls ?\*?**

Bs\_Prak\_1.odt ltext3.txt text04.txt text1.txt text2.txt text3.txt

**Gibt alle Dateien in dem Verzeichnis Bs\_Prak aus.**

linux-8b19:~/Bs\_Prak # **ls text?\***

text04.txt text1.txt text2.txt text3.txt

**Gibt alle Dateien mit den Namen Text aus.**

linux-8b19:~/Bs\_Prak # **ls text[12]\***

text1.txt text2.txt

linux-8b19:~/Bs\_Prak # **ls text[13]\***

text1.txt text3.txt

**Gibt die Text-Files 1 und 2 aus.**

**Gibt die Text-Files 1 und 3 aus. → Damit Unterschied deutlich wird.**

linux-8b19:~/Bs\_Prak # **ls text[1-3]\***

text1.txt text2.txt text3.txt

**Gibt alle Texte aus von 1 bis 3.**

**Wirkung der Zeichen \$?:** Das gesuchte Element endet hierbei mit dem entsprechenden gegebenen Parametern (zB $fs)

ls /etc/ | grep t$ *(als Beispiel alle Files mit einem t am Ende)*

ConsoleKit

PackageKit

aclocal\_dirlist

auto.net

bindresvport.blacklist

default

environment

gdbinit

issue.net

opt

pythonstart

wpa\_supplicant

**Wirkung der Zeichen ^?:** Das gesuchte Element beginnt hierbei mit dem entsprechenden gegebenen Parametern (zB ^fs)

# ls /etc/ | grep ^t

termcap

tightvncserver.conf

tmpdirs.d

tmpfiles.d

ttytype

**Wirkung der Zeichen “\<“:** Bedeutet, dass grep den Wortanfang sucht. Die Anführungszeichen sind hier notwendig, da sonst der Ausdruck nicht ausgewertet wird.

linux-8b19:~ # ls /etc/ | grep "\<t"

mime.types

openmpi-totalview.tcl

termcap

tightvncserver.conf

tmpdirs.d

tmpfiles.d

ttytype

**Geben Sie alle Prozesse auf, die mit k beginnen:** pgrep –fl “^k“

Ausgabe aller Dateien unabhängig von der Position des gesuchten Strings (in dem Fall ein Buchstabe). Für die Ausgabe der Prozesse haben wir zwei verschiedene Befehle herausgefunden. Baumstruktur Darstellung.

linux-8b19:~/Bs\_Prak # pstree | grep "\<k"

|-kactivitymanage---6\*[{kactivitymanage}]

|-kded4---4\*[{kded4}]

|-kdeinit4-+-2\*[dolphin---2\*[{dolphin}]]

| |-2\*[kio\_file]

| |-klauncher

| |-ksmserver-+-kwin---2\*[{kwin}]

| | `-{ksmserver}

|-kdm-+-Xorg

| `-kdm---startkde-+-gpg-agent

| `-kwrapper4

|-kglobalaccel

|-klipper

|-kmix---{kmix}

|-knotify4---{knotify4}

|-konsole-+-bash-+-grep

| `-{konsole}

|-krunner---{krunner}

|-kuiserver

|-kwalletd

|-plasma-desktop-+-ksysguardd

|-polkit-kde-auth---{polkit-kde-auth}

**Was ist der Unterschied zwischen einer Shell- und einer Umgebungsvariablen (environment variable)?:** Die Environment-Variablen werden bei der Prozessgenerierung vererbt, das heißt; die Umgebungsvariablen werden weitergegeben. Da liegt der Unterschied.

**Welche Information enthält die Umgebungsvariable $HOME?:** Pfad des Homeverzeichnis.

**Welche Information enthält die Umgebungsvariable $PATH?:** Die Verzeichnisse, in denen bei einem Programmaufruf nach entsprechendem Programm durchsucht wird.

**Welche Information enthält die Umgebungsvariable $UID:** Die UserID

**Welche Information enthält die Umgebungsvariable $USER:** Name vom eingeloggten Nutzer

**Was bewirkt der Befehl cd \$HOME? Gibt es eine einfachere Alternative?:** Wechselt in das Homeverzeichnis und ja; cd oder cd\.

**Welche Funktion hat die TAB-Rechts Taste bei der Eingabe eines nicht vollständigen Dateinamens oder eines nicht vollständigen Programmna- mens?:** Vervollständig soweit den Dateinamen oder Programm.

**Welche Funktionen haben die Tasten <Pfeil-oben> und <Pfeil-unter>, wenn noch kein Befehl eingegeben wurde?:** Man kann damit die letzten Eingaben durchblättern.

**Welche Funktion hat der history Befehl**?: Listet die bereits ausgeführten Befehle auf.

**Was ist die Funktion der .bashrc Datei im Verzeichnis\$HOME?:** Die .bashrc ist eine Konfigurationsdatei, die beim Starten einer Shell aufgerufen wird und liegt ebenfalls im Homeverzeichnis womit Anpassungen möglich sind.

**Modifizieren Sie die Umgebungsvariable PATH so, dass ein Programm zu- erst im aktuellen Verzeichnis gesucht wird:** PATH=$(pwd):$PATH

**Was macht das Programm (nachzusehen bei der Aufgabe\_1.pdf):** Das Shell-Skript fragt dem jeweiligen Nutzer nach seinem Namen, um diesen auf dem Terminal mit einer Begrüßung auszugeben. Die Funktion ask\_for\_name() fordert eine Benutzereingabe, wo der Nutzername eingelesen wird. Das Skript erlaubt nur die Optin –h | --help. Wenn keine Option angegeben wird, dann wird immer nach dem Nutzernamen gefragt. Dementsprechend wird dann ebenfalls die Begrüßung ausgegeben.