Objektorientierte Programmierung

Anhang A: Kurze Einführung in Linux

Prof. Dr. Stefan Brass

Martin-Luther-Universität Halle-Wittenberg

Wintersemester 2018/19

http://www.informatik.uni-halle.de/~brass/oop18/

Inhalt

Terminal-Fenster, Shell (1)

- Man braucht zunächst ein “Terminal”-Fenster.

  Man muss bei Ubuntu Linux auf “Applications”/“Anwendungen” links oben klicken, dann die Maus auf “Accessories”/“Zubehör” bewegen, und in dem sich dann öffnenden Submenü auf “Terminal” klicken.
  Wenn das Menü zu lang ist, erscheint unten bzw. oben ein Pfeil:
  Bewegt man die Maus darüber, wird der angezeigte Menü-Ausschnitt in der jeweiligen Richtung bewegt.

- Die Eingabeaufforderung (“Prompt”) im Terminal-Fenster hat die Form “ `Benutzer@Rechner:Verzeichnis$` ”.

  Der Prompt ist konfigurierbar, könnte also auch anders aussehen.
  Das Home-Verzeichnis, in dem man sich zunächst befindet, wird mit “`~`” markiert. Der Prompt bedeutet, dass der Kommandointerpreter (“Shell”) bereit ist, ein Kommando entgegenzunehmen.

Terminal-Fenster, Shell (2)

- Bei der Eingabe eines Kommandos im Terminal-Fenster kann man mit
  - ⟨Backspace⟩ das jeweils letzte Zeichen löschen,
  - ⟨Crtl⟩+U die ganze Eingabe,
  - ←, → den Cursor in der Zeile bewegen,

    Normale Zeichen werden dann eingefügt, ⟨Backspace⟩ löscht das Zeichen links vom Cursor, ⟨Delete⟩ das Zeichen unter dem Cursor.
  - ⟨Crtl⟩+C ein laufendes Programm abbrechen oder mindestens unterbrechen.
- Man beendet einen Befehl mit ⟨Return⟩/⟨Enter⟩/↵ .

  Er wird dann ausgeführt.

Terminal-Fenster, Shell (3)

- Das erste Wort auf der Kommandozeile ist der Name des Programms, das ausgeführt werden soll, der Rest sind seine Argumente (Eingabewerte).

  Die Argumente werden also durch Leerzeichen getrennt.

- Ein "ungefährliches" Kommando zum Testen ist "`echo`". Es gibt nur seine Argumente aus.

  `echo Hallo` *(gibt "Hallo" aus)*

- Mit den Cursortasten ↑, ↓ kann man die letzten Kommandos in die Kommandozeile bekommen.

  Und dann erneut ausführen oder ggf. vorher modifizieren.

Terminal-Fenster, Shell (4)

- Letztes Kommando wiederholen:
  ```
  !!
  ```
- Zuletzt ausgeführtes Kommando, das mit "e" beginnt, wiederholen:
  ```
  !e
  ```
- Letztes Kommando wiederholen, aber Zeichenfolge "Hallo" durch "Tschues" ersetzen:
  ```
  ^Hallo^Tschues
  ```

  Es wird nur das erste Vorkommen von "`Hallo`" ersetzt.

Inhalt

Verzeichnisse (1)

- Es empfiehlt sich für diese Vorlesung ein Unterverzeichnis (z.B. "`oop`") anzulegen, das geht mit:

  `mkdir oop` *("make directory")*

- Man wechselt in dieses Unterverzeichnis mit:

  `cd oop` *("change directory")*

- Groß-/Kleinschreibung ist wichtig unter Linux!

- Eventuell legt man hierin noch ein weiteres Unterverzeichnis für das spezielle Programm an (`hello`), und wechselt dann da hinein.

Verzeichnisse (2)

- Nun muss man einen Editor aufrufen, um den Programmtext einzugeben, z.B. mit

  ```
  gedit hello.java
  ```

  Statt `gedit` sind viele andere Editoren möglich: `vi`, `gvim`, `emacs`, `pico`/`nano`.

- Geben Sie das Beispielprogramm ein und speichern Sie ab ("Save").

  Wenn der Editor ein eigenes Fenster hat, brauchen Sie ihn nicht unbedingt zu beenden. Läuft der Editor im Terminal-Fenster, können Sie auch ein weiteres Terminal-Fenster aufmachen, um darin die folgenden Kommandos einzugeben (dort zuerst `cd oop/hello`).
  Sie aktivieren ein Fenster (es bekommt den "keyboard focus", d.h. Tastatureingaben landen dort) durch Mausklick in das Fenster.
  Fenster verschieben: Linke Maustaste über Kopfzeile drücken, Maus verschieben, loslassen.

Verzeichnisse (3)

- Inhalt des aktuellen Verzeichnisses anzeigen:

  `ls` *("list directory")*

  Es sollte nun `hello.java` geben, ggf. auch `hello.java~` (ältere Version).

- Mehr Informationen bekommt man mit

  `ls -l` *("ls, long form")*

  Die Ausgabe ist z.B.

  ```
  insgesamt 1
  -rw-r--r-- 1 brass pib 155 2010-09-17 16:45 hello.java
  ```

  Die erste Zeile besagt, wie viele Blöcke (zu 1KB) die Dateien insgesamt belegen. Dann kommt eine Zeile pro Datei mit Dateityp, Zugriffsrechten, Datum und Uhrzeit der letzten Änderung und anderen Informationen.
  Ein "`d`" in der ersten Spalte markiert Unterverzeichnisse ("directory").
  Der genaue Aufbau wird auf Folie 29 erläutert.

Verzeichnisse (4)

- Alle Dateien anzeigen (auch "versteckte"):

  `ls -al` *("ls, all entries, long form")*

  Dateien und Verzeichnisse, die mit einem Punkt "`.`" beginnen, werden normalerweise nicht angezeigt. Im Homeverzeichnis gibt es davon viele (z.B. üblich zur Speicherung persönlicher Voreinstellungen u.s.w.). Das aktuelle Verzeichnis wird als "`.`" ausgegeben, und das übergeordnete Verzeichnis als "`..`". Wenn man nicht die lange Form des Anzeige wünscht, kann man natürlich auch "`ls -a`" verwenden.

- Dateien im Verzeichnis `hello` listen:

  ```
  ls -l hello
  ```

  Ohne `-l` würden wieder nur die Dateinamen gelistet. Wünscht man Informationen über das Verzeichnis, aber nicht den Inhalt, kann man `-d` ("directory") verwenden.

Verzeichnisse (5)

Wildcards:

- Das Zeichen `*` passt auf eine beliebige Folge beliebiger Zeichen (auch "."), das Zeichen `?` auf ein einzelnes beliebiges Zeichen.
- Alle Java-Dateien auflisten:

```
ls -l *.java
```

- Die "Wildcards" `*` und `?` funktionieren mit beliebigen Befehlen, nicht nur `ls`.

  Der Kommandointerpreter (die "Shell") ersetzt sie vor Aufruf des eigentlichen Kommandos. Die meisten Kommandos erlauben mehr als ein Argument, und führen dann einfach die gleiche Aktion mit allen Argumenten durch (sonst würden die Wildcards nicht funktionieren).

Verzeichnisse (6)

- Name des aktuellen Verzeichnisses anzeigen:

  `pwd` *(“print working directory”)*

- In das übergeordnete Verzeichnis wechseln:

  `cd ..` *(“change directory”)*

- In das Home-Verzeichnis wechseln:

  `cd` *(“change directory”)*

- Platzbelegung (in 1 KByte Einheiten) anzeigen (dies listet auch alle Unterverzeichnisse auf):

  `du` *(“disk usage”)*

Inhalt

Dateien kopieren

- Datei kopieren:

  `cp hello.java x.java` *(“copy”)*

- Datei in anderes Verzeichnis kopieren:

  `cp hello.java hello` *(“cp Dateien Verzeichnis”)*

  Hier wird “`hello.java`” in das Unterverzeichnis “`hello`” kopiert.

- Datei in das aktuelle Verzeichnis kopieren:

  `cp hello/hello.java .`

  Das aktuelle Verzeichnis kann immer mit “`.`” angesprochen werden.

- Verzeichnis mit Inhalt kopieren:

  `cp -r hello test` *(“copy recursive”)*

Dateien umbenennen/verschieben

- Datei (oder Verzeichnis) umbenennen:

  `mv x.java test.java` *("move")*

  Gab es vorher schon eine Datei "`test.java`", wird sie (die alte Version) gelöscht (überschrieben). Das kann auch beim cp-Befehl passieren.
  Dateinamen mit Leerzeichen (oder Spezialzeichen wie "$") sind in `'...'` einzuschliessen: `mv 'Ein Versuch.java' v.java`.
  Dies gilt nicht nur für "`mv`", sondern für beliebige Kommandos (Leerzeichen trennen sonst die Argumente/Eingabewerte eines Kommandos).

- Datei/Verzeichnis verschieben (hier ins übergeordnete Verzeichnis):

  `mv hello.java ..` *("move")*

Dateien löschen

- Datei löschen:

  `rm hello.java` *(“remove”)*

  Gelöschte/überschriebene Dateien sind weg!

  Es empfiehlt sich, Dateien regelmäßig zu sichern (z.B. auf USB-Stick).

- Leeres Verzeichnis löschen:

  `rmdir hello` *(“remove directory”)*

- Verzeichnis mit Inhalt löschen:

  `rm -r hello` *(“remove recursive”)*

- Löschen aller Objektdateien mit Nachfrage:

  `rm -i *.o` *(“remove interactive”)*

Java-Programm compilieren und ausführen

- Wenn man das Java-Programm in der Datei `hello.java` gespeichert hat, kann man es so compilieren:

  `javac hello.java` *("java compiler")*

- Wenn man sich bei der Eingabe des Programms vertippt hat, gibt es Fehlermeldungen.

  Siehe Vorlesung: Kapitel 1 (Langfassung) und Kapitel 2.

- Wenn man alles richtig gemacht hat, wird eine Datei `hello.class` mit dem Java Bytecode für die JVM erzeugt.

  Nutzen Sie `ls`, um das zu überprüfen.

- Das Programm kann dann ausgeführt werden mit:

  `java hello` *("JVM Code ausführen")*

  Beachten Sie, dass man die Endung "`.class`" hier nicht angeben darf.

Inhalt

Graphische Benutzeroberfläche (1)

- Es gibt natürlich auch ein Werkzeug mit graphischer Benutzeroberfläche für die Dateiverwaltung ("Nautilus" File Browser).

  Er ist zu erreichen über das Menü "Places" (dann "Home Folder") oben links.

- Den "Firefox" Web Browser erhält man durch Klick auf die Weltkugel in der Menüleiste oben oder durch Eingabe von "`firefox`" im Terminalfenster.

- Die Möglichkeit zum Ausloggen oder Herunterfahren des Rechners erhält man durch Klick auf das Ein/Aus Symbol oben rechts.

Benutzung eines USB-Sticks

- Steckt man einen USB-Speicherstick in einen Rechner unter Ubuntu Linux, wird er automatisch unter `/media` in das Dateisystem eingefügt.

  Der Name des Verzeichnisses in `/media` ist der Name des USB-Sticks (oft kryptisch). Der USB-Stick wird auch links oben auf dem Bildschirm angezeigt, durch Doppelklick darauf öffnet man ihn in Nautilus. Durch Klick mit der rechten Maustaste öffnet sich ein Pop-up Menü, dort gibt es auch einen Punkt "Safely Remove Drive", den man anklicken sollte, bevor man den USB-Stick herauszieht. Man kann statt dessen auch "`umount /media/*`" eingeben. Auf den "Thin Clients" funktionieren USB-Sticks unter Linux bisher nicht (sie sind nicht vom Server aus zugreifbar). Unter Windows geht es.
  Wenn Sie die Zeilenenden einer Datei zwischen UNIX und Windows konvertieren wollen, können Sie das mit "`fromdos hello.java`" und "`todos hello.java`" machen.

Fenster in Ubuntu Linux (1)

- Genauer: Fenster beim GNOME-Desktop.
- Wenn mehrere Fenster offen sind, ist eins das aktive Fenster (in dem Tastatureingaben verarbeitet werden), zu erkennen an den hellen/farbigen Symbolen in der Kopfzeile.

  Durch Mausklick in ein Fenster (oder auf seine Kopfleiste) wird das das aktive Fenster.

- Man schließt ein Fenster durch Klick auf das Kreuz im orangen Kreis links oben.

  Das zugehörige Programm wird dann beendet. Gleiche Funktion wie das Kreuz rechts oben in der Kopfleiste von Windows-Fenstern.

Fenster in Ubuntu Linux (2)

- Klickt man auf den Pfeil nach unten (oben links in jedem Fenster), wird das Fenster "minimiert".

  Es wird dann nicht mehr im Hauptbereich des Bildschirms dargestellt, sondern nur noch in der Fensterliste unten. Klickt man dort unten, wird es wieder normal dargestellt. Klickt man auf den Eintrag des Fensters unten, während es oben zu sehen ist, wird es auch minimiert. All das funktioniert wie bei Windows.

- Klickt man auf den Pfeil nach oben (oben links in jedem Fenster), wird das Fenster maximiert, es füllt dann den ganzen Bildschirm aus.

  Das Symbol in der Kopfzeile ändert sich dann in ein Quadrat: Klickt man darauf, erhält das Fenster wieder seine alte Größe und Position.

Fenster in Ubuntu Linux (3)

- Man kann Fenster wie in Windows verschieben.

  Die linke Maustaste über der Kopfzeile drücken, Maus verschieben, Maustaste loslassen.

- Fenster können in ihrer Größe verändert werden.

  Auch wie in Windows: Linke Maustaste am rechten oder unteren Rand (oder in der rechten unteren Ecke) drücken (wenn sich der Mauszeiger entsprechend geändert hat), ziehen und loslassen.

- Auch der Scrollbar funktioniert wie in Windows.

  Mit dem Scrollbar am rechten Fensterrand kann man den dargestellten Ausschnitt des Textes verschieben. Man drückt die linke Maustaste über dem Schieber, verschiebt die Maus nach oben oder unten, und läßt los. Alternativen: oberhalb oder unterhalb des Schiebers klicken, oder auf die kleinen Pfeile am Ende, oder Drehrad in der Maus.

Fenster in Ubuntu Linux (4)

- Wenn man ein Textstück markiert, indem man die Maustaste am Anfang drückt und mit der Maus zum Ende fährt, und dann loslässt, kommt es in die Zwischenablage (“Copy”).

  Man braucht also nicht wie bei Windows ⟨Crtl⟩-C zu drücken.

- Durch Klick mit der mittleren Maustaste kann man es wieder einfügen (“Paste”).
- Durch Klick mit der rechten Maustaste erreicht man üblicherweise ein Pop-Up Menü.
- [https://help.ubuntu.com/stable/ubuntu-help/]

Inhalt

Zugriffsrechte (1)

- UNIX/Linux ist grundsätzlich ein System, das auf Kooperation der Nutzer ausgelegt ist.
- Sie können sich z.B. das Verzeichnis mit dem "Hello World" Beispiel des Professors anzeigen lassen:

```
ls ~brass/oop18/hello
```

- Allgemein kann man das Homeverzeichnis des Benutzers *B* mit ~*B* ansprechen.
- Sie können sich auch das Beispiel kopieren:

```
cp ~brass/oop18/hello/hello.java .
```

Zugriffsrechte (2)

- Was genau geht, hängt an den Zugriffsrechten, die der jeweilige Benutzer vergeben hat.
- Z.B. erhält man hier eine Fehlermeldung:

```
ls ~brass/oop18/exam
/home/brass/oop18/exam: Permission denied
```

- Man kann normalerweise auch keine Dateien fremder Benutzer löschen oder verändern.

  Bei ungünstiger (zu liberaler) Setzung der Zugriffsrechte wäre das aber grundsätzlich möglich. Manchmal möchten ja auch einige Freunde/Kollegen zusammen an den gleichen Dateien arbeiten.

Zugriffsrechte (3)

- Die Zugriffsrechte werden mit `ls -l` angezeigt:

```
-rw-r--r-- 1 brass pib   176 2010-09-22 19:06 hello.java
-rwxr-xr-x 1 brass pib 54936 2010-09-22 19:06 prog
```

- Das erste "`-`" ist der Dateityp (normale Datei). Bei Verzeichnissen wird hier "`d`" angezeigt.
- Dann kommen Dreiergruppen "`rwx`" (read, write, execute) von Zugriffsrechten für den Dateibesitzer, seine Gruppe, und alle anderen Rechner-Nutzer.

  Read: Leserechte, Write: Schreibrechte, Execute: Ausführungsrechte.
  Es folgen die Anzahl der "Hard Links" (Einträge dieser Datei im Verzeichnisbaum), der Dateibesitzer, seine Gruppe, die Dateigröße in Byte, Datum und Uhrzeit der letzten Änderung, und der Dateiname.

Zugriffsrechte (4)

- Die Zugriffsrechte für die Datei `hello.java` sind `rw- r-- r--` :
  - Der Besitzer (`brass`) darf die Datei lesen und schreiben (ausführen geht nicht).
  - Die Mitglieder seiner Gruppe sowie alle anderen Nutzer des Rechners dürfen die Datei nur lesen.
- Die Rechte für das Programm `prog` sind `rwx r-x r-x`, d.h. alle dürfen das Programm ausführen.

  Das Programm läuft mit den Rechten des Aufrufers. Selbst wenn das Programm auf Dateien schreiben würde, könnten andere Benutzer damit nicht den Zugriffsschutz für Dateien von `brass` umgehen.

Zugriffsrechte (5)

- Bei Verzeichnissen bedeutet das `x`-Bit, dass man in Pfaden durch das Verzeichnis wechseln darf.

  Wenn man keine Leserechte, aber Ausführungsrechte für ein Verzeichnis hat, kann man es sich nicht anzeigen (auflisten) lassen, aber man kann auf Dateien und Unterverzeichnisse zugreifen, falls man für sie Zugriffsrechte hat, und weiss, wie sie heissen.

- Sie können Zugriffsrechte mit dem Befehl `chmod` ("change mode") setzen, z.B. der Gruppe (`g`) und aller Welt (`o`, "others") die Leserechte nehmen:

  ```
  chmod go-r hello.java
  ```

  Sie selbst wären `u` ("user"), alle zusammen `a` ("all").

  Statt `+` geht auch `-` und `=`.

Zugriffsrechte (6)

- Wir würden es schätzen, wenn Sie dem Kopieren Ihrer Hausaufgaben entgegen wirken, indem Sie das Verzeichnis sperren, z.B.

```
chmod go= oop
```

  Das bedeutet: gar keine Rechte für die Gruppe und alle anderen.

- Wenn Sie das nicht machen, ist das Risiko, dass Sie 0 Punkte bekommen, weil jemand anders Ihre Hausaufgabe kopiert hat.

  Wir prüfen nicht, was Original und was Kopie ist, und halten zumindest aktive Weitergabe zwecks kopieren für nicht in Ordnung.

- Administratoren sind nicht an die Rechte gebunden.

Inhalt

Mehr Informationen (1)

- Man kann Handbuchseiten zu einem Kommando (z.B. `mkdir`) abfragen mit

  `man mkdir` *(“manual”)*

  Mit ⟨`Enter`⟩ kommt man eine Zeile im Text weiter, mit ⟨`Leertaste`⟩ eine Seite, mit “`b`” (“back”) eine Seite zurück, und mit “`q`” (“quit”) kann man die Anzeige der Handbuchseite verlassen. Das Handbuch besteht aus mehreren Bänden, z.B. stehen Systemaufrufe im zweiten Band: “`man 2 mkdir`” sucht `mkdir` in Band 2.

- Viele Kommandos geben eine Kurzanleitung aus, wenn sie mit der Option `--help` aufgerufen werden:

  `mkdir --help` *(Hilfe zu* `mkdir`*)*

Mehr Informationen (2)

- Nach Stichworten im Handbuch sucht man mit

  ```
  apropos mkdir
  ```

  Alternativ auch mit "`man -k mkdir`".

- GNU-Software ist häufig durch `info`-Seiten besser beschrieben, als durch den `man`-Eintrag:

  ```
  info mkdir
  ```

  Info ist ein Hypertextsystem, das ohne Maus auskommt: `n` führt zu nächsten Seite, `p` zur vorherigen, `u` eine Stufe hoch, `/` zur letzten besuchten Seiten und wenn der Cursor über einem mit "`*`" markierten Hyperlink steht, folgt man ihm durch Drücken von ⟨`Enter`⟩.

- Siehe: [http://www.ee.surrey.ac.uk/Teaching/Unix/]
  [http://www.ernstlx.com/linux90bash.html]
  [https://www.ubuntu.com]