Einstieg in Software

- Einstieg in Technologie und Programmierung erleichtern
- HowtoEncourageWomenToLinux
- Tipps zum Programmieren lernen
- Unix Tutorial
  - Arbeiten mit der Shell / wie bekomme ich Hilfe zu Kommandos
  - Gemütlich machen: was beim Login eingestellt werden kann und was das Arbeiten erleichtert
  - Weitere wichtige Kommandos und Abläufe

Einstieg in Technologie und Programmierung erleichtern

Was kannst Du tun, um anderen Menschen (mit Fokus auf FLINTA) die bisher nicht viel damit zu tun hatten, den Einstieg in die Programmierung und den Umgang mit Technologie zu erleichtern?

Nachfolgend eine Übersicht verschiedener Angebote, Medien und Artikel:

Online Angebote: Do Stuff

Angebot	Fokus	Kommentar
khanacademy.org	Einstieg in unterschiedliche Felder der Computerwissenschaften (EN, Anmeldung erforderlich)	Algorithmen, Kryptografie, Informationstheorie, Wie funktionieren Computer, Internet 101
geekgurldiaries.co.uk	"collection of video logs about using and making technology, along side interviews with inspirational women in the fields of computing, science, technology and engineering"	Raspberry Pi, information and communications technology, Carrie Anne Philbin
blockly-games.appspot.com		
code.org	Einstieg in die Programmierung	Minecraft, Star Wars
scratch.mit.edu	Blockprogrammierung; hier bis jetzt nur mit erwachsener Begleitung mit einem konkreten Vorhaben	
snap.berkeley.edu		selbes Prinzip wie Scratch, aber deutlich mächtiger - braucht aber auch jemand zum Erklären am Anfang
coderdojo.org	enthält Anleitungen und Resourcen, um selbst ein CoderDojo aufzuziehen bzw. um im Selbststudium eine Programmiersprache zu lernen (Ruby, Scratch, Python, Wearables, JavaScript,...)	Inhalte je nach Sprache unterschiedlich

Chaos macht Schule	Workshops und Materialien zu den Themen:
	`Datenschutz`
	`Medienkompetenz`
	`Soziale Netzwerke`
	`Wie funktioniert das Internet?`
	`Das Urheberrecht in der digitalen Welt`	

girlswhocode.com		
arduino.cc		
projects.raspberrypi.org	Projekte und Bildungsmaterialien zu Rasperrypi	
kobakant.at	Super Resource für e-textile Ideen und Projekte	

Gruppen

Angebot	Fokus	Kommentar
Women who Go (https://www.womenwhogo.org/)	Golang Meetup/Slack Gruppe	weltweit, in DE bisher nur in Berlin
Rails Girls Berlin (http://railsgirlsberlin.de/)	Beginner Kurse und Project Groups	
Heart of Code (http://heartofcode.org/)	FLINTA Hackerspace (Python, Dataviz, Ruby on Rails,...)	donnerstags Damensalon, andere Termine: http://heartofcode.org/kalender
NASA Datanauts (https://open.nasa.gov/explore/datanauts/)	online community fokussiert auf NASA Open Data	tolle internationale Gruppe von Data Nerds (>90% Frauen), die mit Open NASA Daten tinkern
Girls in Tech Berlin (https://www.meetup.com/de-DE/Girls-in-Tech-Berlin/?_cookie-check=K4cGpGy2cb_2Ox5-)		

Literatur

Titel	Autorin	Kommentar
Good Night Stories for Rebel Girls		tolle Illustrationen, Geschichten mittelmäßig

Ada Lovelace und der erste Computer	Fiona Robinso	tolle Illustrationen, Geschichte mittelmäßig
Einfach Programmieren für Kinder (Carlsen Clever)	Diana Knodel, Philipp Knodel, Jan Radermacher	Buch und App
Circuits	http://thesoftcircuiteer.net/wp-content/uploads/2018/01/CraftingAPath_etextile_zines_vol-1.pdf	Zine zu eTextile Projekte/alter 4+
Disobedient electronics	http://www.disobedientelectronics.com/resources/Hertz-Disobedient-Electronics-Protest-201801081332c.pdf	Zine disobbedient electronics
Riot Girl Manifesto	https://rarlindseysmash.com/posts/riot-code-grrrl-manifesto	Zine Manifesto
The Thrilling Adventures of Lovelace and Babbage: The (Mostly) True Story of the First Computer	Sydney Padua	Buch, sehr gut recherchiert mit gute informationen

HowtoEncourageWomenToLinux

(Notiz: letztes Update dieser Seite war 2007, weiter unten aufgeführte Links zur haecksen.org Seite sind defekt)

HOWTO: Wie ermutigt man Frauen, sich mit Computern auseinanderzusetzen

Zusammenfassung

Dieser Artikel erklärt ein paar Schwierigkeiten und Vorurteile, mit denen sich Frauen in der Computerszene auseinandersetzen müssen und untersucht problemlösende Strategien, mit dem Ziel, dass sich mehr Frauen in die Computerszene einbringen.

Einleitung
Auf dem Linux Symposium in Ottawa (Kanada) in 2002 moderierte ich eine 'Linux Chix: Birds Of a Feather Session' (BOF). Während dieser Veranstaltung und der gesamten Konferenz hörte ich immer wieder dieselben Fragen:

"Meine Freundin hasst Windows. Wie kann ich sie dazu kriegen, Linux zu benutzen?"
"In meiner lokalen Computergruppe sind fast keine Frauen. Wie kann ich das ändern?"
"Warum gibt es nicht mehr Frauen in der Open Source Entwicklung?"
Ganz offensichtlich würden die Leute in der Computerszene es begrüssen, wenn sich mehr Frauen aktiv mit Computern auseinandersetzten - allerdings wissen sie nicht, wie sie das beeinflussen können und was überhaupt die Gründe für sowenig Engagement auf Frauenseite sind. Dieses HOWTO nimmt die Empfehlungen, Erklärungen und Vorschläge der Frauen auf, die sich bereits in der Computerszene engagieren. Dieses Dokument nahm seinen Anfang in den mündlichen äusserungen der LinuxChix auf der BOF, und wurde in den darauf folgenden Monaten immer wieder erweitert und ergänzt. Mit anderen Worten drückt dieses HOWTO die Gefühle und Meinungen von echten Frauen aus, die sich mit Computern beschäftigen. Obwohl wir diejenigen Frauen repräsentieren, die es 'geschafft haben', wissen wir trotzdem relativ genau, warum andere Frauen das Computerumfeld verlassen, oder sich dort niemals eingebracht haben. Wir sind uns ausserdem des Druckes bewusst, der uns momentan aus der Gemeinschaft aussondert.

Wir möchten hier besprechen, warum sich Frauen generell von Computern fernhalten, warum sie sich speziell von Linux fernhalten und was man tun kann, um Frauen zu ermutigen, sich doch damit auseinanderzusetzen. Wir hoffen, dass dieses HOWTO hilft, dass sich mehr Frauen in der Computerszene wohlfühlen und sich dort aktiv einbringen.

Publikum

Unser Hauptpublikum sind männliche Computer Enthusiasten, die gerne mehr Frauen in der Computerszene involviert sehen würden. Wir möchten aber auch Menschen ansprechen, die bisher zu beschäftigt mit ihrem Lieblingshobby waren, um darüber nachzudenken, warum die meisten Frauen ihre Interessen nicht teilen. Wir hoffen, dass nach dem Lesen dieses HOWTOS ein Eindruck entsteht, warum Frauen sich von Computern fernhalten, und ein wir ein paar Ideen rübergebracht haben, was man tun kann, um diese Tendenz aufzuhalten.
Wir sprechen nicht die Leute an, denen die Nichtanwesenheit von Frauen egal ist, oder die denken, dass Frauen nicht an den Computer gehören. Wenn du nicht der Meinung bist, dass Frauen aufgrund von externen Einflüssen von Computern verdrängt werden, dann wird dich dieses HOWTO nicht vom Gegenteil überzeugen. Es wird allerdings ein paar Anhaltspunkte geben, bei denen man weiter recherchieren kann.
Wir wollen definitiv keinem Geek dabei helfen, weibliche Geeks zu finden, mit denen er sich dann verabreden kann. Es ist ein zentraler Widerspruch beim Thema Frauen und Computern, dass die Männer, die am lautesten nach mehr Frauen im Computerbereich schreien, auch die sind, die sie am ehesten vergraulen. Geeks, die in allen computerinteressierten Frauen nur potentielle Freundinnen sehen, verhalten sich oftmals in einer Weise, die Frauen abschreckt. Wir möchten erklären, welche Verhaltensweisen Frauen abschrecken, und welche sie ermutigen.

Welches Problem? Sexismus gibt es nicht mehr!

Ein Argument, das ich ständig höre, ist folgendes: 'Welches Problem? Es gibt doch gar kein Problem. Sexismus ist ausgestorben! Frauen halten sich von Computern fern, weil sie das so möchten!' Ich dachte früher auch immer, Sexismus wäre ausgestorben. Ich entdeckte dann, wie falsch ich lag, als ich mich in mehrere 'Frauen und Computer' Mailinglisten eintrug. Woche für Woche posteten die Frauen neue Geschichten darüber, wie sie diskriminiert und beleidigt wurden, weil sie Frauen waren. Diese Geschichten waren keine Jahrzehnte alt, oder handelten von Leuten, die mit Sexismus grossgeworden sind. Es waren ganz alltägliche Geschichten von modernen Frauen, in modernen Situationen. Sie wurden einfach aus ihrem gewählten Arbeitsfeld rausgeekelt. Das ist keine theoretische Annahme - viele Frauen verlassen das Computerumfeld tatsächlich nur aufgrund von unangenehmen Erfahrungen in der Schule oder auf Arbeit, die Vorgesetzte und sexistische Kommentare involvieren.

Die untenstehenden Links stellen mein Lieblingsbeispiel des modernen Sexismus dar:

Ursprüngliches Posting einer Frau auf eine Sydney Linux User Group Mailingliste

Reply posting, das das Problem als 'überstresstes Weib' diagnostizieren

Zweites Reply posting, das dem ersten zustimmt

Die zwei Replies waren dann genug, um sie zu frustrieren

Satirische und heldenhafte Antwort einer anderen Frau

Ein Mann, ders immer noch nicht begriffen hat

Die perfekte Antwort eines Mannes, der es begriffen hat

Sexismus ist überall vorhanden, und er veranlasst Frauen, sich ihm zu entziehen. Man könnte jetzt argumentieren, dass die Leute, die sich über das 'gestresste Weib' lustig machen, unwissend, blöd oder gutmeinend sind, und dass die Postings nicht als 'Sexismus' gesehen werden sollten, aber das Ergebnis ist das Gleiche: Aktionen dieser Gestalt veranlassen Frauen, das Computerumfeld zu verlassen - etwas, das wir gerne verhindern möchten.

Über die Autorin
Val Henson ist eine Linux Kernel Entwicklerin, ein aktives Mitglied der LinuxChix und weiblich. Sie interessiert sich für die theoretische und praktische Entwicklung von Betriebssystemen, für das Thema Frauen und Computer, und für gutes Bier. Viele Frauen halfen ihr, das vorliegende HOWTO zu produzieren, unter anderem:

- Raven Alder
- Suzi Anvin
- Poppy Casper
- Claudia 'Texchanchan' Crowley
- Steph Donovan
- Joy Goodreau
- Telsa Gwynne
- Amy Hieter
- Hanna Linder
- Anna McDonald
- Marcia Barret Nice
- Miriam Rainsford
- Carla Schroder
- Jenn Vesperman
- Jenny Wu
- Megan 'Piglet' Zurawicz
- Safari
- Tina Lorenz (Übersetzung ins Deutsche)
- Nika Bertram (Übersetzung ins Deutsche)
- Daniela Garling (Übersetzung ins Deutsche)
- Und viele andere, die gerne anonym bleiben möchten

Warum gibt es so wenig Frauen in der Computerszene?
Es gibt viele ausgezeichnete Bücher und Studien, die sich mit diesem Thema tiefgehend befasst haben - wir können nur zusammenfassen, warum Frauen sich von Computern fernhalten. Wir möchten ausserdem ein paar der gängigsten Theorien über dieses Thema auf ihren Wahrheitsgehalt hin überprüfen.
Drei gute Quellen für das Thema 'Frauen und Computer sind folgende:

'Unlocking the Clubhouse: Women in Computing' von Jane Margolis und Allan Fisher (englisch)

http://www.amazon.com/exec/obidos/ASIN/0262133989

'Women in Computing Keyword List' http://women.acm.org/search/key_list.php (Nur manche der hier referenzierten Dokumente sind online verfuegbar)
'Why Are There So Few Female Computer Scientists' von Dr. Ellen Spertus

http://www.ai.mit.edu/people/ellens/Gender/pap/pap.html
Wir sollten bei den zwei häufigsten Theorien anfangen, warum es so wenig Frauen in der Computerbranche gibt:

'Frauen interessieren sich einfach nicht für Computer' und 'Frauen sind nicht so schlau wie Männer'. Das Problem mit dem Satz 'Frauen interessieren sich einfach nicht für Computer' ist, dass es eigentlich gar nichts aussagt. Es ist das Gleiche wie die Frage 'Warum ist der Himmel blau?' mit 'Weil der Himmel halt blau ist' zu beantworten. Das Argument dahinter ist, dass Frauen von Geburt an genetisch dazu vorbestimmt sind, sich nicht für Computer zu interessieren. Sehr wenig Leute sind bereit, das wirklich in sovielen Worten zu sagen, aber das ist die eigentliche Nachricht hinter dem 'weil es halt so ist' Argument. Wenn man jetzt das genetisch bedingte Desinteresse ausschliesst (was niemandem so wirklich schwer fallen sollte), muss eruiert werden, welche umgebungsbedingten Einflüsse involviert sind.
Eine mehr ins Detail gehende Version dieser Theorie ist 'Frauen sind nicht so schlau wie Männer' oder etwas in der Art - Frauen können bestimmte Aufgaben nicht so gut wie Männer, üblicherweise Mathematik, räumliches oder logisches Denken. 'Newsweek' trumpht regelmässig mit Studien auf, die geschlechtsspezifische Unterschiede gefunden haben wollen, wobei sie ignorieren, dass es viel mehr Studien gibt, die diese Unterschiede genau nicht finden. Häufig können andere Forscher die Ergebnisse beim zweiten Test nicht wiederherstellen, oder finden Fehler in den Methoden des Originalprojektes. Aber diese Geschichten tendieren dazu, bei der Presse auf taube Ohren zu stossen.

[Hier fehlt eine ganze Menge, die ich noch machen muss]

Wie man Frauen ermuntert, und wie man es nicht tut
Um mehr Frauen für die Computerszene zu gewinnen, brauchen wir beides: sowohl Tipps dazu, was man tun, als auch dazu, was man unterlassen sollte. Wir haben diese Sachen hier in "Richtig" und "Falsch"-Paaren zusammengefasst, denn das ist hilfreicher als je eine reine Liste mit Do's oder Don'ts. Einige dieser Ideen wirken vielleicht selbstverständlich, andere sind schwieriger nachzuvollziehen. Man sollte nicht den Fehler machen, diese Vorschläge nicht ernst zu nehmen - dies sind die Ratschläge der echten Frauen, die an die Terminals gelockt werden sollen. Viele dieser Vorschläge sind jedoch auch geschlechts-unspezifisch, lassen sich also auch dazu nutzen, die Computerszene für alle anderen Leute attraktiver zu machen.

Falsch: Sexistische Witze erzählen
Dies ist in Wirklichkeit ein grösseres Problem, als den meisten bewusst ist. Ich habe mehr als einmal gehört, wie ein Mann erklärte, er würde diese Art von Witzen nicht machen, und dann, Stunden oder nur Minuten später, hörte ich denselben Mann einen Witz zum Besten geben über schwangere Frauen oder PMS. Vermutlich glauben sie, es sei in Ordnung, Witze zu machen über

etwas, das "wahr" oder "witzig" ist. Dabei merken diese Menschen nicht, dass - trotz aller Rechtfertigungen, "Wahrheitsbekundungen" und Witzeleien - diese sexistischen Witze gleich jeder Sorte fast immer abwertend und diskriminierend für Frauen sind, und sie extrem wütend machen, unabhängig vom jeweiligen Zusammenhang. Es ist egal, ob man seinem Frauen-sind-doof-Witz einen Männer-sind-doof-Witz voranstellt, es bleibt verletzend und abschreckend für die Frauen im Publikum. Blondinenwitze sind nicht witzig - sie machen sich nicht über dumme Blonde lustig, sondern über dumme Frauen.
Natürlich kann man dagegen halten, dass Frauen nicht so empfindlich sein sollen (ich würde allerdings nicht zustimmen), aber selbst dann, egal, ob zu sensibel oder nicht, vertreiben diese Kommentare und schlechten Witze die Frauen. Falls du dir nicht sicher bist, ob dein Witz Frauen beleidigen könnte, nimm am besten an, dass er genau das tut, und denk dir was anderes aus. Man sollte daran denken, dass wir in der Realität leben - und dass "sollte" eine Sache, und "wird" eine andere ist. Es ist falsch, Dinge zu tun, die Frauen vertreiben, auch wenn es nicht so gemeint ist.

Richtig: Protestieren, wenn jemand sexistische Witze erzählt
Beim nächsten Mal, wenn du jemanden dabei erwischst, wie er sexistische Witze über Frauen auf deiner Mailingliste oder leibhaftig reisst, sag ihm deine Meinung. Natürlich trägt man dabei das Risiko, sich selbst zum Zielobjekt zu machen, aber für Frauen ist es noch schwerer, hier einzugreifen. Frauen schweigen oft, wenn sie mit sexistischen Witzen konfrontiert werden, denn würden sie protestieren, würde man sie sofort angreifen und als "über-sensibel" bezeichnen, unentspannt oder gar "Kampfemanze" nennen. (Anm.d.Ü.: Im englischen Original stand hier das Wort 'Feminazi'. Unseres Erachtens nach wird das in Deutschland nicht benutzt. Obwohl 'Kampfemanze' eine deutlich andere Konnotation hat als 'Feminazi', deshalb fuer die eins zu eins Übersetzung nicht taugt, finden wir alle Begriffe dieser Art gleich rei&szlib;erisch und diskreditierend.)

Am besten lassen sich sexistische Witze mit Humor bekämpfen. Wenn jemand zu einer Mail über die technischen Skills einer Frau schreibt "Ist sie noch Single?", sollte man mit, "Wow, Michael, kein Wunder, dass DU noch Single bist." antworten. Jedes Mal, wenn eine Frau eine sexistische Bemerkung oder Witzelei sieht, fühlt sie sich wütend, ausgeschlossen und klein gemacht. Jedes Mal, wenn eine Frau sieht, wie ein Mann sich gegen solch ein Verhalten stellt, fühlt sie sich angenommen - und es macht sie stolz.

Falsch: Frauen 'Groupies' oder 'Schlampen' nennen

Das Wort "Schlampe" zu benutzen wirkt diskriminierend für alle Frauen, egal, auf wen es sich bezieht. Ich dachte schon, es sei nicht nötig, dies extra zu erwähnen, aber offensichtlich ist diese Einsicht keine Selbstverständlichkeit. Denn erst neulich hörte ich, wie einige Linux-Entwickler den Begriff "Schlampe" in vollem Ernst und demonstrativer Coolness verwendet haben. Nein, jemanden eine "Schlampe" zu nennen, macht dich nicht cool. Hör nicht auf die Gerüchte. Das Gleiche gilt für das Wort 'Groupie'. Es spielt keine Rolle, ob das Wort deiner Meinung nach zutrifft oder nicht, keine Frau möchte so genannt werden. Jede Frau im Computerumfeld (und jede andere auch) ist mehr wert, als dass man sie auf eine Funktion oder ein Stück Fleisch reduzieren darf.

Richtig: Ein wenig Respekt zeigen

Nicht nur den Frauen gegenüber, die du attraktiv findest, sondern gegenüber allen Frauen, ebenso wie allen anderen Erscheinungsformen menschlichen Lebens. Damit zeigst du, dass du Frauen und andere als menschliche Wesen wahrnimmst, dass deine emotionale Reife den Entwicklungsgrad eines Dreizehnjährigen überschritten hat, und dass eine Frau mit dir keine Zeit verschwenden würde.

Falsch: Jemandem die Tastatur aus der Hand nehmen

Dies ist ein generelles Problem, wenn jemand etwas Neues lernen soll, aber es passiert häufiger Frauen. Jemand stellt eine Frage, und anstatt dass du dieser Person eine Antwort gibst, nimmst du ihr die Tastatur weg und gibst den Kommandobefehl selbst ein. Das ist falsch! Es macht das Lernen schwerer, und die betreffende Person fühlt sich dumm und hilflos.

Richtig: Klare Erklärungen und Richtungsweisungen geben

Obwohl es schwerer ist, sich die Zeit und Geduld zu nehmen, um zu erklären, was genau man tut und warum, um dann den Kommandobefehl zu buchstabieren, wird sich der Aufwand langfristig lohnen, denn hierbei lernt die andere Person und es ist sehr unwahrscheinlich, dass man die gleiche Frage noch einmal beantworten muss. Grade Frauen werden sich sehr viel sicherer in ihren Fähigkeiten fühlen, wenn man es ihnen ermöglicht, die Kommandos selbst einzugeben.

Falsch: überstürzte Annäherungsversuche machen

Stell dir eine Kneipe oder einen Pub vor voller Sportfanatiker, Fans eines Spiels, das du nicht kennst oder kennen willst. Stell dir vor, sie wären alle grösser und stärker als du, würden in einer Sprache sprechen, die du nur halb verstehst, und jeden belächeln, der sich nicht mit Leib und Seele ihrem Sport verschrieben hat. Nun stell dir vor, du würdest diese Bar betreten, mit einem T-Shirt mit dem Aufdruck, "Ich hasse Sport." Stell es dir einfach mal vor. Wie würdest du dich fühlen? Nervös? ängstlich? Fremd? Fehl am Platz?

Dann hast du schon eine klitzekleine Vorstellung davon, wie sich eine einzelne Frau inmitten einer riesigen Männergemeinschaft fühlt.

Denk dir dieses Gefühl von Nervosität mit, wenn du den Rest dieses Abschnitts liest. Wenn du eine Frau auf einem Linux User Group Treffen oder online anbaggerst, gibst du ihr das Gefühl, dass sie nicht dazu gehört, als wäre sie im Kreuzfeuer, und riskiere geächtet zu werden, wenn sie dir einen Korb gibt oder sich wehrt. Denk daran, dies ist keine nette Begegnung unter vier Augen, wo sie sich gut dabei fühlt, dir die Wahrheit zu sagen, sie ist umgeben von einer Menge, die den zuvor genannten Sportfans vergleichbar ist. Sie möchte dazu gehören, ein Teil der Gruppe werden, und indem du ihr diesen Schlag versetztst, nimmst du sie aus der Herde und isolierst sie von der Gruppe. Frauen wachsen auf in dem Bewusstsein konstanter Angst davor, von Männern angegriffen zu werden, und - so albern es auch klingen mag, davon sind all ihre Handlungen geprägt, egal, wie sicher oder weltoffen sie auf Männer wirken mag.

Wie jeder andere Mensch auch, wünscht sich eine Frau, Freunde zu haben und so geschätzt zu werden wie sie ist. Jedes Mal, wenn sie eine Email erhält mit einer Einladung zu einem Date, wird sie daran erinnert, dass sie nicht als Teil der Gruppe gesehen wird, sondern als etwas anderes, als Sexualobjekt, und dass sie mit Sicherheit nicht allein an ihren technischen Fähigkeiten gemessen wird.

Die Erkenntnis mag für dich schwer verdaulich sein, doch du solltest einsehen, dass es keinen Sinn

macht, Frauen auf Treffen der Computerszene anzumachen. Mit grosser Wahrscheinlichkeit wirst du damit NICHT deine einzige Chance auf die grosse Liebe verspielen, indem du sie nicht direkt überfällst, sondern stattdessen eher die Chance wegwerfen, ein neues, witziges Mitglied der Computerszene hinzuzugewinnen. Und selbst wenn du noch glauben solltest, die Chance zu deiner grossen Liebe zu verpassen, denk daran, dass die meisten Frauen, die tapfer genug sind, auf einem Treffen der LUG oder deiner Mailingliste aufzutauchen, sowieso in der Regel die Frauen sind, die den ersten Schritt machen. Indem du bei ihr gleich mit der Tür ins Haus fällst, vertreibst du sie, und mit ihr ebenso all die anderen Frauen, die vielleicht auch interessiert gewesen wären, wenn die erste Frau geblieben wäre.

Dies gilt umso stärker für Frauen, die du per Email oder im IRC kennen lernst. Du denkst vielleicht, deine "Bist du noch Single?"-Frage ist wahnsinnig gewitzt und gerissen, aber sie hat es schon tausendmal gehört. Auch wenn du es nur witzig meinst, oder schon eine Freundin hast, oder gar verheiratet bist - tu's einfach nicht.

Richtig: Freundlich sein
Wenn Frauen nicht angebaggert werden, werden sie anstelle dessen häufig komplett ignoriert. Das ist nicht wirklich besser. Frauen, die neu zu einer Gruppe dazukommen, wollen meistens dasselbe, was Männer wollen - sich willkommen fühlen, über gemeinsame Interessen sprechen und Freunde gewinnen. Wenn eine Frau etwas sagt, hör ihr zu und antworte auf eine freundliche Art. Fange an, dich mit ihr zu unterhalten und finde ein Thema, an dem ihr beide interessiert seid. Nimm nicht an, dass sie 'typisch weibliche' Interessen oder Meinungen hat, nur weil sie eine Frau ist. Versuch stattdessen, frei von Vorurteilen zu bleiben und herauszufinden, wofür sie sich interessiert. Höchstwahrscheinlich hat sie Interessen, die über Make-Up, Haare und Filmstars hinausgehen, da sie in die Computerszene involviert ist.
Einige Frauen haben sich beschwert, dass alle Männer scheinbar nur darüber mit ihnen reden, wie wenig Frauen es im Computerumfeld gibt. Obwohl das ein wichtiges Thema ist, würden es viele Frauen vorziehen, über etwas anderes zu sprechen. Ganz speziell möchten sie nicht daran erinnert werden, wie seltsam sie erscheinen, wenn sie das erste Mal auf einem Computer-Treffen auftauchen. Warte, bis sie sich in der Gruppe niedergelassen hat, und sich wohlfühlt; dann kannst du sie zu dem Thema befragen, wenn es dich wirklich interessiert.

Falsch: Sich über zuwenig Frauen im Computerumfeld beschweren
Es ist nützlich und konstruktiv, über das Fehlen von Frauen in der Computerszene zu sprechen, wenn man sich vom Standpunkt der Frauen an das Thema annähert, die bei einem spannenden und belohnendem Feld aussen vor sind. Es ist traurig und armselig, über das Fehlen von Frauen im Computerumfeld zu sprechen, wenn man sich auf einen Standpunkt von Männern bezieht, die ihr nichtvorhandenes Liebesleben auf diesen Umstand zurückführen. Der beste Weg, Frauen zu nerven und sie zu vergraulen ist der, in dieser Art über die fehlenden Frauen zu sprechen. Untenstehend ein paar der üblichsten Reaktionen von Frauen, die Männern dabei zuhören, wie sie über den Mangel an Frauen jammern:

- Was bin ich, unsichtbar? Weiss er überhaupt, dass ich hier bin?
- Gut zu wissen, dass ich anscheinend nur existiere, um einsamen Männern zu dienen
- Armselig. Ihr seid so armselig.
- Warum tust du dann nichts dagegen, anstatt dich zu beschweren?

- Und wieder einmal nehmen alle an, dass nur Männer zuhören
- Vielleicht sollte ich gar nicht in diesem Umfeld sein
- Was ist mit mir kaputt, dass ich hier bin, aber andere Frauen nicht?
- Er bezieht alles auf sich selbst
- Kein Wunder, dass der keine Freundin hat
- Ich bin nicht nur in einer Fleischerei, nein, ich bin auch noch das Schnitzel!

Wie man sehen kann, führt das Geheule nur dazu, dass die Frauen genervt sind oder sogar gehen und nicht mehr wiederkommen. In keinem Fall führt es dazu, dass mehr Frauen gewillt sind, mit dir auszugehen.

Richtig: Frauen ermuntern, sich Computern zu nähern
Anstatt sich nur über zuwenig Frauen zu beschweren, tu was, um diesen Zustand zu ändern. Nimm die Beschwerden der Frauen ernst (fang mit diesem HOWTO an), lies die Studien darüber, warum Frauen sich von Computern, Mathe oder Wissenschaft im allgemeinen fernhalten und finde heraus, wie du Frauen am besten motivieren kannst. Sei ermutigend und unterstützend, wenn andere Leute adrüber diskutieren, warum es so wenige Frauen in der Computerszene gibt. Wenn sich die Gelegenheit ergibt, versuche einer Frau als Mentor zu nutzen. Mentoren sind Menschen, die jemanden in Bildung und Karriere anleiten, ermutigen und beraten. Nicht jeder ist dazu erschaffen, Menor zu sein, und es ist schwierig, passende Mentoren und Schüler zusammenzuführen, aber wenn es klappt, können die Resultate spektakulär sein. Menro sein ist allerdings kein Weg, eine Freundin zu finden - alles, was ein Mentor aus der Beziehung bekommt, ist der reflektierte Ruhm seines Schülers und die Freude, eine andere Person wachsen zu sehen.

Falsch: Neue Frauen wie Aliens mit vier Köpfen anstarren
Niemand mag es, wenn er angestarrt wird. Warum sollten also Frauen es mögen? viele Frauen beschweren sich darüber, dass wenn sie in einen Raum voller Geeks laufen, plötzlich jedwede Unterhaltung aufhört, alle sich umdrehen und glotzen, und ein paar sogar mit dem Finger auf sie zeigen. Das ist einschüchternd und unschön, und mehr als genug, um eine Frau schwören zu lassen, niemals wiederzukommen.
Ein gutes Zitat von Mia, einem LinuxChick: "Zu einem Linux-User-Treffen zu gehen, hat mich nie interessiert, aber ich war auf einer Reihe von anderen Geekveranstaltungen, wo sich jeder umdrehte und starrte, als ich den Raum betrat. Es fühlte sich mehr an wie die 'Fremder in der Bar' Szene in einem Western, als alles andere."

Richtig: Neuankömmlinge höflich behandeln
Wenn Frauen auf einem Computertreffen auftauchen, oder auf Mailinglisten posten, verhalte dich neutral. Versuche, sie wie jede andere Person zu behandeln, die du in deiner Gruppe gerne dabei haben würdest. Denk dran, es ist für sie kein Kompliment, wenn du sie fortlaufend daran erinnerst, dass sie einzigartig ist, selten oder einfach nur seltsam. Wenn du anfängst dir vorzustellen, dass Frauen ein normaler Teil der Computerszene sind, könnte das sehr schnell Wirklichkeit werden.

Falsch: Frauen ihrem Klischee entsprechend behandeln
Nimm nicht an, dass alle Frauen gerne kochen, nähen und Babies toll finden, und dass sie nur aktiv in der Computerszene sind, weil sich ihr Freund/Mann/Sohn dafür interessiert. Eine Frau sagte, dass

wann immer jemand auf Computer-Treffen ihr etwas erklären wollte, er Analogien zur Küche oder zu Babies herstellte, weil er annahm, dies seien Themen, mit denen sie am vertrautesten sei. Denk nicht, dass sich Frauen nicht auch für Autos, Mathe, Kampfjets oder Roboter interessieren. Denk nicht, sie könnten nicht auch ihre Kernel selbst kompilieren - persönlich kenne ich 15 Frauen, die das können, und einige von ihnen schreiben selbst Kernelcode. Wenn du Glück hast, wird eine davon auf deinem nächsten Geektreff oder auf deiner Computermailingliste auftauchen, und du möchtest sie nicht beleidigen, indem du annimmst, sie könne noch nichtmal ihren eigenen Rechner installieren. Denk nicht, dass sie mit der Computerei angefangen hat, weil sie so gerne gechattet hat. Übrigens: Frauen schimpfen genausohäfig wie Männer. Nimm dich also nicht extra zurück, wenn du vor einer Frau fluchst. Wenn sie schon mal Linux Kernel Code gelesen hat (insbesondere arch/sparc/), dann hat sie das Wort 'fuck' zumindest schonmal gelesen.

Richtig: Frauen wie normale Menschen behandeln

Behandle die Frauen in deiner Gruppe wie normale Menschen, denn sie sind es. Einige Leute beschweren sich: 'Frauen wollen wie normale Leute behandelt werden, aber erzählen mir gleich danach, ich soll keine sexistischen Witze reissen! Das ist doch paradox!' Nun ja, wenn du 'normale Leute' als 'die Männer, mit denen ich normalerweise abhänge' definierst, dann ist das ein Paradoxon. Wenn du aber Frauen in die Definition von 'normale Leute' mit einschliesst, und dann normale leute in einer fairen und respektvollen Art behandelst, dann brauchen Frauen auch keine Spezialbehandlung.

Wenn du dir immer noch nicht sicher bist, wie du Frauen behandeln sollst, versuch das Folgende: Sei freundlich, aber nicht überfordernd, sei leger, fange ein Gespräch an, wie du es normalerweise tust, bleib nicht an ihr kleben, wenn das Gespräch vorbei ist. Wenn du die meiste Zeit mit einer sehr spezifischen männlichen Untergruppe verbringst, wirst du dein Verhalten in mancherlei Hinsicht ändern müssen. Aber dies würde auch zutreffen, träfest du einen Mann mit einem völlig anderem Horizont. Wenn du feststellst, dass du dein Verhalten komplett umstellen musst, um Frauen nicht zu verärgern, solltest du dir überlegen, es sowieso und in allen Gelegenheiten umzustellen. Keiner wird dir glauben, wenn du einfach aufhörst, in der Gegenwart von Frauen sexistische Bemerkungen zu machen, sie aber weiter machst, wenn (anscheinend) keine Frauen in der Nähe sind.

Falsch: Zuviel Kritik

Frauen sind so sozialisiert, dass sie auf Kritik deutlich sensibler reagieren als Männer, und sie sind auch deutlich selbstkritischer. Als Ergebnis lassen sich Frauen viel eher von unfairer oder harter Kritik vergraulen als Männer. Wenn du versucht bist, zu kritisieren, denk an das Folgende: Genau gar keiner wurde mit der Fähigkeit zum programmieren oder Kernel kompilieren geboren, und irgendwann wusstest auch du noch nichts über Computer. Leute verlieren das Interesse an einer Sache, in der sie sich als schlecht begreifen. wenn du also jemanden dazu bringen möchtest, sich weiterhin für Computer zu interessieren, versuch diese Person nicht soviel zu kritisieren, dass sie glaubt, sie wäre schlecht darin.

Lob an der richtigen Stelle

Frauen haben im Durchschnitt ein sehr viel niedrigeres Selbstvertrauen als Männer und betrachten sich generell selbst viel kritischer als jeder Aussenstehende. Lob an der richtigen Stelle hilft, ihr Selbstvertrauen aufzubauen, wodurch sie sich wiederum weiter für das Thema interessiert. Wenn

sie glaubt, sie könnte nicht gut mit Computern umgehen, wird sie wahrscheinlich damit aufhören.

Das folgende sind einige Tips für Lob an der richtigen Stelle:

- ○ Sei aufrichtig und ehrlich. Wenn du in Wahrheit denkst, ihr Programm ist ein hässliches Stück Muell, dann sage ihr nicht, dass du dessen syntaktische Schönheit bewunderst. Finde etwas, dass du wirklich bewundern kannst und lobe sie dafür.

- ○ Sei spezifisch. "Du kannst gut mit Computern umgehen" bedeutet gar nichts, "Du weisst immer, welche Linux-Distribution zu empfehlen ist" ist spezifisch und deshalb bedeutsam.

- ○ Mache dein Lob passend. Lobe eine Linux Kernel-Entwicklerin nicht dafür, dass sie Linux installieren kann. Lobe eine GIMP Entwicklerin nicht fürr ihren Umgang mit Layern. Dein Lob soll signifikante Fertigkeiten reflektieren anstatt deine Ignoranz ihrer Expertise.

- ○ Vergleiche mit dir selbst. Wenn sie bash-scripting schneller gelernt hat als du, sag ihr das. Sag "Wow, du hast bash-scripting in X Monaten gelernt. Ich hab 2*X Monate gebraucht, um das zu lernen." Oder wenn sie einen dummen Kompilierungsfehler gemacht hat, erzähl ihr von deinem schlimmsten Kompilierungsfehler. Wenn sie mitbekommt, dass ihre Fehler nicht ungewöhnlich sind, wird sie sich besser fühlen.

- ○ Lobe erst, bevor du kritisierst. Wenn du konstruktive Kritik anbringen willst, ist es eine gute Idee, ihr erst einmal zu sagen was sie richtig gemacht hat.

- ○ Lobe sie und kritisiere nicht. Nicht immer auf ein Lob eine Kritik folgen lassen. Lobe sie stattdessen öfter und belasse es dann dabei.

- Gib nicht an. Aus dem nichts heraus zu sagen "Sie kann ihren eigenen Kernel kompilieren!" ist nicht schmeichelhaft, es ist eher angeberisch, als hättest du etwas mit ihrem Erfolg zu tun oder wärst in irgendeiner Weise dafür verantwortlich. Eltern sind besonders anfällig fuer Angebereien. Unterschwellige, subtile Hinweise auf ihr Können sind viel besser - "Oh, naja, wenn du eine Frage ueber Kernel Kompilierung hast, wird sie

dir wahrscheinlich eher weiterhelfen können als ich." Wenn jemand auf meine Fähigkeiten auf diese Art aufmerksam macht, ist das unbeschreiblich schön.

Du solltest ihr auf keinen Fall Komplimente über ihre Haare, ihr Gesicht, ihren Körper oder ihrer netten Art machen. Wenn sie sich für Computer interessiert ist sie, per Definition, ein Geek und möchte vermutlich für ihre Intelligenz, ihre Fähigkeiten und ihre harte Arbeit gelobt werden. Lobe ihre erste Linux-Installation oder für ihre intelligenten und interessanten Fragen beim letzten Treffen. Ein Kompliment für irgend etwas anderes ist unangebracht und wird als sexuelle Annäherung verstanden (weil es das meistens auch ist) und sie wird sich weniger wohl und weniger selbstsicher fühlen.

Falsch: Nur Männer als Referenten zu Computertreffen einladen
Wenn alle eure Redner Männer sind, werden Frauen das bemerken und sich nicht willkommen fühlen. Vorbilder, mit denen man sich identifizieren kann sind wichtig, um das Interesse für etwas aufrecht zu erhalten.
Richtig: Auch Frauen einladen
Es ist erstaunlich einfach, technisch kompetente Frauen zu finden, die bereit sind, zu eurer Gruppe zu kommen und einen Vortrag zu halten. Weibliche Redner sind wahrscheinlich der effizienteste Weg, Frauen für eure Veranstaltung zu gewinnen. Sie werden Gelegenheit haben, ein Vorbild zu sehen, und sich wenigstens für ein paar Stunden nicht so fühlen, als wären sie die einzige Frau, die sich für Computer interessiert.

Falsch: Aber ich mach das doch gar nicht!
Jetzt ist wahrscheinlich ein guter Zeitpunkt, sich mal selbst zu beobachten. Bei der LinuxChix BOF auf dem Ottawa Linux Symposium hatten wir die Liste mit Gründen fertig, aus denen Frauen sich von Computern fernhalten. Ein Mann aus der örtlichen Linux-User-Group meldete sich und sagte, dass niemand in seiner Gruppe irgendwelche der Dinge täte, ueber die wir uns beschwert hatten, dennoch hätten sie Probleme, Frauen anzuziehen. Eine Frau aus der selben Gruppe meldete sich und sagte: "Sie tun es doch." Sie sagte weiterhin, dass nur einige "schwarze Schafe" sich so verhielten, aber diese wenigen seien Grund genug, die meisten Frauen abzuschrecken. Das ist ein wichtiger Punkt: Wenn deine Gruppe neun hilfsbereite und höfliche Mitglieder hat, und ein ungehobeltes, sexistisches, lautes Mitglied, werden die meisten Frauen wegen dieses einen Mitglieds weiter fernbleiben. Ich sehe ein, dass das den anderen Mitgliedern der Gruppe gegenueber unfair ist, aber so ist nun mal die Realität. Wenn es in deiner Gruppe ein solches schwarzes Schaf gibt, versucht es mit ein wenig Gruppenzwang, wenn er das nächste Mal etwas tut, das Frauen abschreckt. Antwortet auf seine e-Mail, widersprecht dem, was er sagt -- stellt klar, dass ihr seine Ansichten nicht teilt. Zu wissen, dass wenigstens eine andere Person willens ist, dem schwarzen Schaf öffentlich zu widersprechen wird immens helfen und wird dafür sorgen, dass Frauen gewillter sind zu bleiben.
Ich habe selbst immer und immer wieder Männer sagen hören, dass sie diese Dinge nicht tun würden und sah sie dann Stunden oder Minuten später das tun, von dem sie zuvor behauptet hatten, es nie zu tun. Ich denke nicht, dass irgendeiner dieser Männer gelogen hat, es war ihnen einfach nur nicht bewusst. Sexistische Witze oder Kommentare zu machen, scheint das am weitesten verbreitete unbewusste Verhalten zu sein -- viele Männer erkennen einfach nicht, dass das, was sie sagen, beleidigend auf Frauen wirkt.

Ausserdem ist es auf jeden Fall möglich, gute Absichten zu haben und dennoch Frauen abzuschrecken. Vielleicht denkst du eine Frau zu ermutigen, indem du ihr zu ihrem Mut gratulierst, bei einem bestimmten Event aufzutauchen. In Wirklichkeit verdeutlichst du ihr aber, dass sie eigenartig und anders ist, anstatt ihr das Gefuehl zu geben, sie wäre ein Teil der Gemeinschaft. Eine Frau formulierte es so: "Ich weiss, ich bin eine Ausserirdische. Du brauchst es nicht extra zu betonen." Wir hoffen, dass dieses HOWTO dir helfen wird, zu erkennen, wenn du unbewusst Frauen abschreckst.

Wenn du neugierig bist, wie dein Verhalten auf Frauen wirkt, kann ich nur empfehlen, dass du zu einer Bekannten gehst, die zu direkter Offenheit neigt, und sie fragst, ob sie sich an etwaiges offensives Verhalten von dir gegenüber Frauen erinnert. Vergiss nicht, die meisten Frauen würden sich eher einen Arm ausreissen als einem Mann gegenüber unfreundlich zu sein, also kann es schwer sein, eine ehrliche Antwort zu bekommen.

A. LinuxChix

LinuxChix ist eine aktive und wachsende Organisation, die von und für Frauen, die sich für Linux interessieren, geführt wird. Gegründet von Deb Richardson und derzeit geleitet von Jenn Vesperman, spezialisiert sich LinuxChix darauf, eine unterstützende und freundliche Umgebung für alle Linux Nutzer und Entwickler zu bieten, aber insbesondere für Frauen. LinuxChix wird von einer internationalen Gruppe von Freiwilligen geführt, die glauben, dass es wichtig ist, Frauen in die Linux Community zu integrieren. Männer dürfen LinuxChix beitreten, aber Frauen stehen im Mittelpunkt und wir versuchen, eine Umgebung beizubehalten, in der Frauen dominieren. Zu den Frauen, die derzeit bei LinuxChix involviert sind, gehören verschiedene Linux Kernel Entwicklerinnen, eine Mozilla Entwicklerin, ein Mitglied der GNOME Foundation, eine O'Reilly Autorin, System Administratorinnen, Computer Consultants, Sicherheitsexpertinnen, Schülerinnen und Studentinnen vom High School Level bis hin zu Doktoren in vielen Bereichen, buchstäblich hunderte Programmiererinnen verschiedenster Art und viele Computerhobbyistinnen. Wenn du eine Frau bist, die sich für Linux interessiert, oder du kennst eine solche, ist LinuxChix ein ausgezeichneter Ort, Gleichgesinnte zu finden.

Kürzlich hat LinuxChix eine Reihe von Features, Web-Ressourcen und Mailinglisten ihrer Webseite hinzugefügt. Wenn du sie vor mehr als einigen Monaten mal besucht hast, solltest du einen zweiten Blick drauf werfen. Die neuen Features beinhalten Linux- Kernel-Hacking-Kurse, verschiedene neue Mailinglisten, Online- Programmierkurse, Buch- und Softwarerezensionen und vieles vieles mehr. Eine Menge LinuxChix Regionalkreise wurden vor kurzem angefangen oder wiederbegonnen (Regionalkreise gibt es, um face-to-face Treffen zwischen LinuxChix zu ermöglichen). Der LinuxChix Entwicklungsprozess ist offen und freundlich. Wir begrüssen neue Freiwillige und Ideen, subscribe einfach die Mailinglisten und biete deine Hilfe an.

Auf unserer Website kannst du mehr über LinuxChix erfahren:

http://www.linuxchix.org

Tipps zum Programmieren lernen

Online (Self-Study)

z.B.

- https://learn.freecodecamp.org - ausführlicher online Kurs zu HTML, CSS und jede Menge Javascript
- https://www.codecademy.com/catalog/subject/all - bietet einige kostenlose Kurse für verschiedene Sprachen an (aber auch kostenpflichtige).
- https://www.codewars.com/, kleine Coding-Challenges, Schwierigkeitslevel steigert sich je mehr man löst, für viele verschiedene Sprachen

Empfehlung! Was man zum Programmieren sonst noch braucht - Command Line, Editors, git:

- https://www.learnenough.com/command-line-tutorial
- https://www.learnenough.com/text-editor-tutorial#sec-modern_text_editors
- https://www.learnenough.com/git-tutorial

Für Berliner*innen

- Im feministischen Hackspace http://heartofcode.org/ gibt es coole Gruppen, u.a. Python beginner learning group, IT security, Data visualisation.
- http://railsgirlsberlin.de/ bieten ab und zu Anfängerinnenwochenenden, sowie regelmäßige Projektgruppen und online-Ressourcen zum Lernen von Ruby on Rails an. Der Name ändert sich gerade, weil "girls" ein bisschen missverständlich ist - Zielgruppe sind vor allem erwachsene Frauen. Ruby ist eine anfängerfreundliche Programmiersprache, die z.B. im Bereich Webentwicklung zum Einsatz kommt.
- https://www.womenwhocode.com/berlin - monatliche Talk-nights, in den anderen Wochen Hack-Nights vorrangig, aber nicht nur für Frauen. Eher für Leute, die mindestens ein

bisschen programmieren können - glaube ich.

Unix Tutorial

Letzte Änderung war Dez. 2018. Tutorial entstand als Teil der Adminenfibel, by Princess.
Ich gehe hier vor "was brauche ich zuerst". Eine strukturierte Anleitung zur Bash findet sich z.B. unter https://www.linuxwiki.de/Bash

Arbeiten mit der Shell / wie bekomme ich Hilfe zu Kommandos

Für die remote administration eines Unix/Linux-Servers ist es unerläßlich, die Kommandozeile (Shell) zu beherrschen, weil eine grafische Nutzeroberfläche (GUI) meist nicht zur Verfügung steht.

Die **Standard-Shell unter Linux ist die bash** (Bourne-Again-Shell), die sich aus der Bourne Shell (sh) entwickelt hat.

Die Admine loggt per ssh auf many.haecksen.org mit ihrem Benutzeraccount (nicht gleich als root) ein. Idealerweise ist ein ssh-Schlüssel dort hinterlegt, so daß die Passworteingabe entfällt. Gibt es keinen ssh-Schlüssel, wird das Passwort abgefragt. Dieses Verhalten ist im sshd (dazu später) konfiguriert.

Einmal auf der Shell "gelandet", ist es gut, zu wissen, daß es drei Arten von Kommandos gibt. Gibt man einen Befehl ein, so wird in folgender Reihenfolge nach dem Kommando gesucht:

- ist es ein **Shell-Alias** (also ein meist selbst konfigurierte Befehlsabkürzung oder -Abfolge)
- ist das Kommando ein **Shell-Builtin**
- oder ist das **Kommando ein Programm "im Pfad"**.

Je nachdem unterscheidet sich, wie man sich Hilfe holt:

- Bei eigenen Aliasen gibt es keine Hilfe, es sei denn, diese wurde explizit implementiert (was sich bei eigendefinierten Abkürzungen meinst nicht lohnt)
- Bei Shell-Builtins bekommt man Hilfe mit **"help kommando"**
- Bei normalen Kommandos im Pfad gibt es in der Regel eine Manpage: **"man kommando"**

type / wie finde ich heraus, was ich denn für ein Kommando vor mir habe

Um herauszufinden, was wir vor uns haben, tippen wir:

```
type kommandoname
```

oder auch

```
type -a kommandoname
```

um alle Vorkommen eines Kommandos anzuzeigen.

Beispiel:

```
type cd
cd is a shell builtin
```

Hier bekommen wir also Hilfe mit

```
help cd
```

Beispiel 2:

```
type -a ls
```

ergibt bei mir (Princess):

```
ls is aliased to `ls -F'
ls is /bin/ls
```

Das erste ls ist also von mir selber definiert, keine Hilfe. Das zweite ist aber ein Kommando "im Pfad", also mit Angabe eines Verzeichnisses (/bin). Hier erhalte ich also Hilfe mit

```
man ls
```

Aufgabe: diese Seite anlesen. "durch" ist sportlich.

Der Pfad / die PATH Variable

Was aber hat es genau mit "dem Pfad" auf sich: **$PATH** ist eine **Shell-Variable**. Anzeigen läßt man sich den Inhalt mit:

```
echo $PATH
```

darauf folgt eine Angabe wie z.B. diese:

```
/home/princess/bin:/client/bin:/client/sbin:/usr/local/bin:/bin:/usr/bin:/sbin:/usr/sbin:/usr/X11R6/bin:.
```

Wir sehen hier eine **Abfolge von Verzeichnissen, getrennt mit dem Doppelpunkt**. In dieser Reihenfolge wird in den Verzeichnissen nach dem gerade eingegebenen Kommando gesucht, wenn es eben nicht von der Shell kommt. Wichtiger Hinweis: am Ende befindet sich der Punkt, also das aktuelle Verzeichnis. Dieses ist in einer root-Shell in der Regel NICHT gesetzt, weil es das Ausführen von untergeschobenen Kommandos erleichtern würde.

Festgelegt wird diese Variable entweder im systemweiten Konfigurationsfile /etc/bash.bashrc oder im Konfigurationsfile im Homeverzeichnis des Nutzers.

Das Homeverzeichnis des Nutzers liegt auch in einer Shell-Variable: $HOME.

Aufgabe: gib dieses aus.

In einem eigenen Verzeichnis /home/princess/bin kann ich eigene Kommandos und Skripte ablegen.

Die Konfigurationsfiles von Kommandos

Typischerweise gibt es zu jedem Unix Kommando ein Konfigurationsfile, das idR. auf "rc" für "resource config" endet. Das systemweite File der bash ist oben erwähnt. Die eigene Konfiguration im Home-Verzeichnis beginnt mit dem Punkt und endet auf rc, also in dem Fall: .bashrc . Der komplette Pfad der Datei lautet also:

```
/home/username/.bashrc
```

Ebenso gibt es z.B. für das Mailprogramm mutt das systemweite /etc/Muttrc und das .muttrc im Home, oder für den Editor vi das systemweite /etc/vim/vimrc und das .vimrc im Home.

Bewegen in Verzeichnissen: cd

Da wir ja schon wissen, daß dies ein Shell-Builtin ist, hole Dir Hilfe zu diesem Kommando.

cd ohne weitere Argumente wechselt in das Home-Verzeichnis des Nutzers. Ein Verzeichnis im Baum "höher" erreicht man mit **"cd .."**. Für das "Springen" in Verzeichnisse gibt es grundsätzlich zwei Wege:

- **absolute** Adressierung mit vollem Pfad
- **relative** Adressierung vom aktuellen Verzeichnis aus

Das Wurzelverzeichnis eines jeden Unix heißt "/" (vorwärtsgerichteter Schrägstrich, gesprochen "root"). Unterverzeichnisse des Systems liegen darunter:

```
bin   dev  home        lib    lost+found  mnt  proc  run   srv  tmp  var
boot  etc  initrd.img  lib64  media       opt  root  sbin  sys  usr  vmlinuz
```

Aufgabe: wie kommt man mit ls zu diesem listing?

Die absolute Adressierung ist unabhängig von dem Verzeichnis, in dem ich mich aktuell befinde. Nach dem Login bin ich in meinem Home, also /home/username. Möchte ich mir eine Datei anzeigen lassen, z.B. die Systemweite COnfig-Datei vom vim, geht das also mit

```
less /etc/vim/vimrc
```

Wenn ich relativ adressiere, ist mein Ausgangspunkt relevant. Das aktuelle Verzeichnis lasse ich mir ausgeben mit **"pwd"** für "print working directory", falls ich unsicher bin, wo ich mich befinde.

Möchte ich nun mein Verzeichnis bin wechseln, reicht:

```
cd bin
```

Möchte ich in das Verzeichnis eines anderen Users gehen (soweit Rechte gesetzt sind), ginge das mit:

```
cd  ../user2
```

also ein Verzeichnis hoch und dann der Name des anderen Users.

Aufgabe: wie hieße der absolute Pfad zum anderen user? Zum ersten Beispiel: wie hieße der relative Pfad zum vimrc vom Homeverzeichnis aus?

---

less / der Pager / Dateien ansehen

Bei den Konfigurationsfiles von Programmen handelt es sich um reine ASCII-(Text)Dateien, die man mit einem Pager ansehen kann. Das ursprüngliche Unix-Kommando dafür hieß "more". Unter Linux hat sich aber das in seinen Möglichkeiten umfangreichere **"less"** durchgesetzt.

Man kann nun

- in das Verzeichnis wechseln und mit **"less dateiname"** eine Datei anzeigen oder
- den vollen Pfad der Datei beim less angeben

Aufgabe: versuche beides für /etc/vim/vimrc .

Was aber passiert, wenn ich versehentlich eine Binärdatei ausgeben lassen will? **Aufgabe:** Teste es mit /bin/ls .

Natürlich kann ich vor dem Ansehen oder dem Versuch, eine Datei zu editieren (auch das geht nur mit Textdateien) herausfinden, was für eine Datei ich vor mir habe. Der Dateityp hängt unter Linux, anders als unter Windows NICHT an der Endung der Datei! Man findet den Dateityp heraus mit

```
file dateiname
```

Auch hier kann relativ oder absolut adressiert werden.

Aufgabe: führe das file-Kommando aus auf /etc/vim/vimrc, /bin/ls, eine Bilddatei, ein libreoffice-Dokument, ein PDF File....

---

mkdir: Verzeichnisse anlegen

Um sein Homeverzeichnis ein wenig zu strukturieren, braucht man **Verzeichnisse**.

Sinnvoll sind das bisher erwähnte Verzeichnis bin (Groß- und Kleinschreibung sind signifikant unter Unix!), sowie ein Verzeichnis tmp.

Diese kann man mit

```
mkdir name
```

anlegen.

Möchte man gleich mehrere Unterverzeichnisse mit anlegen, gibt es die Option -p:

```
mkdir -p Projekte/Adminen
```

Aufgabe: lege ein paar mögliche Projektverzeichnisse in Deinem $HOME an. Hier kann man Namen recht frei gestalten, bei den Systemverzeichnissen (/etc, /usr, /bin, /sbin...) sollte man tunlichst nichts an den Namen ändern.

Namenswahl: welche Zeichen in Namen sind erlaubt?

Im Prinzip: alle außer ASCII Null. Auch Leerzeichen in Dateinamen sind möglich, da man auf der Kommandozeile aber das Leerzeichen braucht, um bei Kommandos Optionen und Argumente zu trennen, **rate ich von Leerzeichen in Dateinamen ab**, außer man möchte sich viel und oft die Finger brechen. (Leer- und andere Sonderzeichen können aber durch \ (backslash) maskiert werden, das macht auch die bash bei der automatischen Dateinamenergänzung mit <tab><tab>).

Unkritische Zeichen sind: Buchstaben a-z-A-z, Umlaute sollten vermieden werden, Punkt, Bindestrich, Unterstrich. Nur als erstes Zeichen sollte der Bindestrich (Minus) vermieden werden, weil damit Optionen zu Kommandos eingeleitet werden und man dann Mühe hat, solche Dateien wieder zu löschen.

Als hilfreich erwiesen haben sich auch Datummstempel für eine Art "Versionierung für Arme": man schreibt das Datum statt in der Form **DD.MM.JJJJ** (was zwar menschenlesbar ist, aber für eine Sortierung mit ls nicht geeignet) in der Form **JJJJMMDD**, evtl. noch mit Zeit: JJJJMMDDSSMMSS, also z.B. 20181206 für Nikolausi 2018.

Bearbeitet man Dateien und möchte im Fehlerfall auf eine frühere Version zurückgreifen können, kopiert man die Datei vor dem Editieren auf eine Version mit Datumsstempel, z.B.

```
cp .bashrc .bashrc.20181206
```

Hat man mehrere Versionen, listet ls diese dann chronologisch, weil es Integer sortiert.

Übrigens kann man **lange, sprechende Namen** wählen: **die bash ergänzt** Namen bei Eindeutigkeit, wenn man die **Tabulator-Taste** verwendet. <tab><tab> listet alle Namen bei Nicht-Eindeutigkeit. Ohne zuvor eingegebenes Kommando listet <tab><tab> alle zur Verfügung stehenden Befehle, fragt aber bei einer größeren Anzahl vorher nach, ob man wirklich alles gelistet haben möchte oder eher ein paar mehr Buchstaben zur Eindeutigkeit angibt).

---

rm: Dateien wieder löschen

Gelegentlich möchte man auch aufräumen und Dateien wieder löschen. Dazu gibt es

```
rm dateiname
```

In dieser Form wird die Datei dann ohne Nachfragen sofort gelöscht. Es empfiehlt sich, ein Alias (s.u.) rmi zu definieren für rm -i. Insbesondere, wenn man mehrere Dateien löschen möchte, wird dann bei jeder nachgefragt und man kann mit y oder n antworten.

Das Gegenteil bewirkt -f für force.

Leere Verzeichnisse löscht man mit

```
rmdir verzeichnisname
```

Sind die Verzeichnisse nicht leer, sollen aber gelöscht werden, verwendet man

```
rm -rf
```

VORSICHT ist geboten, hier sollte sicherheitshalber immer das Verzeichnis, in dem man sich befindet, überprüft werden (mit pwd) oder man läßt sich das aktuelle Verzeichnis immer im Prompt anzeigen (s.u., Einstellungen).

Der Editor: warum man vi(m) können muß

Eigentlich ist Unix/Linux ein Betriebssystem wo sich vieles nach gusto einstellen läßt und jeder sich die Umgebung so einrichten kann, wie es ihr beliebt.

Einige Systemkommandos jedoch rufen standardmäßig den vim als Editor auf. vim ist die neuere, besser zu bediendende Version des vi. Es lohnt sich also, diesen immer verfügbaren Editor als Standard zu erlernen.

Anleitungen finden sich zuhauf, z.B. unter https://www.selflinux.org/selflinux/html/vim.html

vi ist also idR. ein alias auf vim (**Aufgabe:** teste dies mit den bisher gelernten Kommandos!)

An dieser Stelle aber in aller Kürze: es gibt drei Modi des vi, zwischen denen man zu wechseln verstanden haben sollte.

Der vi wird gestartet mit

```
vi dateiname
```

sodann befindet man sich im **Bewegungsmodus**. Man KANN sich im vim zwar mit den Pfeiltasten im Text bewegen, jedoch empfiehlt es sich, die Tasten

```
hjkl
```

für

```
links, unten, oben, rechts
```

zu verwenden, weil diese, kombiniert mit einer vorangestellten Ziffer, das schnelle Springen im Text ermöglichen. Dieser Editor wurde für alte, langsame Terminals entwickelt und ist daher sehr effizient.

Ist man an die Stelle gesprungen, an der man etwas editieren will, kann man in den **Editiermodus** wechseln:

```
i     füge vor dem aktuellen Zeichen ein (insert)
I     füge am Anfang der Zeile ein
a     füge nach dem aktuellen Zeichen ein (append)
A     füge am Ende der Zeile ein
o     eröffne eine neue Zeile unterhalb der aktuellen
O     eröffne eine neue Zeile oberhalb der aktuellen
```

sodann kann Text eingefügt werden.

Löschen:

```
x     löscht das aktuelle Zeichen
dd    löscht die aktuelle Zeile
```

Aufgabe: lese in der Anleitung, die wortweise gesprungen bzw. gelöscht wird. Auch: wie löscht man mehrere Zeilen auf einmal?

Der **Editiermodus** wird beendet mit

```
ESC
```

und man ist wieder im **Bewegungsmodus**.

Zum Speichern (oder Einlesen von Dateien oder suchen und ersetzen) braucht man den Kommandomodus. In diesen wechselt man vom Bewegungsmodus mit : (Doppelpunkt). Darauf folgen Kommandos

```
:w        speichere die Datei ab
:w name   speichere die Datei unter einem neuen Namen ab
:r datei  lese eine Datei ein
:r !kommando lese den Output eines Kommandos ein, z.b. :r !date ist gut für Kommentare
:wq       speichern und verlassen (geht auch mit ZZ)
:q!       verlassen ohne Speichern
```

Gemütlich machen: was beim Login eingestellt werden kann und was das Arbeiten erleichtert

Jetzt, wo wir zumindest in groben Zügen einen Editor beherschen, können wir uns in den Konfigurationsfiles zur bash Einstellungen bei Variablen und Aliasen machen, die das Arbeiten erleichtern können.

Aufgabe: wie hieß nochmal das Konfiguratiosfile der bash?

Variablen setzen beim Login

Bei der bash gibt es eigentlich zwei Files, die ausgeführt werden: .bash_login für Loginshells und .bashrc für andere. Da diese Unterscheidung aber fürs tägliche Arbeiten irrelevant ist, machen wir da keinen Unterschied und schreiben alles, was wir brauchen ins .bashrc. Ins .bash_login schreiben wir nur:

```
source ~/.bashrc
```

damit das .bashrc beim login ausgeführt wird. Selbstverständlich wird vor den eigenen Files das systemweite Configfile ausgeführt, in diesem Fall ist es das /etc/profile (aus historischen Gründen) und /etc/bash.bashrc.

Wir wollen nun Variablen setzen (wie man sie abfragt s.o.!).

Eine Variable wird gesetzt mit

```
NAME=inhalt
```

Sie ist dann aber nur im aktuellen Prozeß (dazu später) verfügbar. Damit sie auch in allen Kindprozessen gilt, brauchen wir hier immer:

```
export NAME=inhalt
```

Dies gilt für die Kommandozeile wie für das .bashrc.

Wichtig: alles, was wir ins .bashrc schreiben, wird erst beim nächsten login ausgeführt. Um das **.bashrc neu einzulesen**, führt man auf der Kommandozeile

```
source .bashrc
```

aus.

Das TMP-Dir

Viele Programme brauchen ein temporäres Verzeichnis, um Dinge ab- oder zwischenspeichern zu können, daher setzen wir zwei Variablen:

```
export TMPDIR=/home/username/tmp
export TMP=/home/username/tmp
```

Stelle sicher, daß das Verzeichnis auch angelegt ist.

Der Prompt / die Eingabeaufforderung

Hier kann man durch geschickte Einstellung sich das Arbeiten sehr erleichtern. Der Prompt wird in der Variable PS1 festgelegt (ja es gibt auch PS2, dies kommt bei einfachen Shellscripten auf der Kommandozeile zum Tragen).

Aufgabe: gib den Inhalt von PS1 aus.

Im Prompt kann man username, hostname, aktuelles Verzeichnis und eine laufende Nummer in der bash-history angeben, daher ist eine sinnvolle Einstellung:

```
export PS1="\u @ \h:\w \!> "
```

Häufig will man sich aber anzeigen lassen, wenn man mit root-Rechten unterwegs ist, daher verwendet man lieber:

```
if [ -w /etc/passwd ]; then
  PS1="\h:\w \!# "
else
  PS1="\u @ \h:\w \!> "
fi
export PS1
```

dann wird als root das Doppelkreuz # statt > angezeigt.

Aufgabe: schreibe das erste Beispiel auf der Shell und sieh, was sich sofort ändert. Schreibe das zweite Beispiel ins .bashrc. Verwende gerne Kommentare zu Deinen Eintragungen: kennzeichne diese mit vorangestellten # (vor jeder Kommentarzeile).

Nimm gerne weitere Änderungen anhand der Anleitung zur bash vor.

Die Editor-Variable

Gibt den Standard-Editor an. Jede Userin kann diesen für ihre eigene Umgebung selber wählen, aber spätestens als root wird es immer der vi. Daher stellen wir ein:

```
EDITOR=/usr/bin/vi;                export EDITOR
VISUAL=/usr/bin/vi;                export VISUAL
```

Hier neu: mit dem Semikolon kann man mehrere Kommandos in einer Zeile schreiben und trennen.

Die Bash-History

Die bash merkt sich die letzten N Kommandos, wenn man es denn einstellt:

```
HISTSIZE=200;                      export HISTSIZE
```

Nach Beenden der Shell werden die Kommandos aus dem Speicher in das File .bash_history geschrieben.

Möchte man nicht, daß ein Kommando in der History landet, schreibt man ein Leerzeichen vor das entsprechende Kommando.

Um Dubletten zu vermeiden, setzen wir außerdem:

```
history_control=ignoredups;                export history_control
set command_oriented_history;              export command_oriented_history;
```

Hacker setzen gern die HISTSIZE=0, damit nicht mitprotokolliert wird, was sie tun.

Aufgabe: da das History File erst beim Beenden der Shell geschrieben wird, was muss ein Hacker genau tun, um das zu verhindern oder seine Spuren zu verwischen?

Die Path-Variable

..hatten wir schon behandelt. Als root möchte man das aktuelle Verzeichnis (".") nicht dabeihaben, deswegen setzen wir:

```
if [ -w /etc/passwd ]; then
   PATH="$HOME/bin:/usr/local/bin:/bin:/usr/bin:/sbin:/usr/sbin:/usr/X11R6/bin"
 else
   PATH="$HOME/bin:/usr/local/bin:/bin:/usr/bin:/sbin:/usr/sbin:/usr/X11R6/bin:."
fi
```

Der Manpath

Ähnlich wie die Pfad-Variable legen wir noch die Pfade fest, wo nach manpages gesucht wird:

```
MANPATH=/usr/man:/usr/local/man:/usr/X11R6/man:/usr/openwin/man:/usr/share/man:/usr/share/catm
an:/usr/catman;       export MANPATH
```

Aufpassen: alles muss in eine Zeile! Kein Umbruch!

Diverse kleine Dinge

```
  export PAGER=less
```

(nur dass das klar ist :))

```
  export LC_COLLATE=C
```

bestimmt die Sortierreihenfolge bei ls: zuerst "unsichtbare Dateien". Diese beginnen mit einem Punkt und werden nur bei ls -a angezeigt, sodann alphabetisch sortiert. Erst dann kommen die regulären Dateien, die mit Buchstaben oder Zahlen beginnen in der Reihenfolge: Ziffern, Großbuchstaben, Kleinbuchstaben (dies ist eine Empfehlung).

Sinnvolle Aliase

IdR. gilt: eigene Abkürzungen und Aliase NICHT so nennen wie das Ursprungskommando. Allerdings gibt es Ausnahmen, die auch nicht dramatisch sind: less ist z.B. nur gut nutzbar, wenn der Inhalt der ausgegebenen Datei nicht beim Beenden von less verschwindet.

Daher:

```
alias less='less -MeiQX'
alias les=less
```

ebenso:

```
alias df="df -h"
```

Aufgabe: finde heraus, was die Optionen im Einzelnen bedeuten

Als weiterhin sinnvoll haben sich ergeben:

```
alias 0="sudo $SHELL"
alias    ls='ls -F'
alias    ..='cd ..'
alias pstree='pstree -A -u'
alias rmi="rm -i"
```

Wieder zu allen Kommandos die **Aufgabe:** finde heraus, was sie tun und was die Optionen heißen.

Funktionen

Auch einfache Funktionen können definiert werden:

```
function ll     () { ls -alFh $* | less -MeiQ ;}
function l      () { echo $1* ;}
```

Über Prozesse sprechen wir noch, und auch über grep:

```
function psg    ()
{
  echo 'UID PID PPID C STIME TTY TIME CMD'
  ps -ef | grep $* | grep -v grep
}
```

Der Phantasie sind kaum Grenzen gesetzt und für komplizierte Kommandos kann man sich immer Abkürzungen bauen.

Das .vimrc

Auf älteren Unices benutzt man hier .exrc, also falls Ihr mal das .vimrc nicht vorfindet.

Auch hier sind mannigfaltige Einstellungen möglich! Startet den vi und ruft

```
:set
```

auf, da wird schon eine Menge angezeigt. Ich erachte als sinnvoll:

```
set autoindent       # Einrückungen in der nexten Zeile übernehmen, fürs Programmieren
set wrapmargin=8     # 8 Zeichen vor Zeilenende in die nächste gehen
set textwidth=72     # Text nicht breiter als 72 Zeichen machen, das ist der kleinste gem.
Nenner
```

Braucht man einmal Zeilen, die länger sind und weil man Textpassagen mit Einrückungen hereinpasten möchte, setzt man im laufenden Editor:

```
:set paste
```

das überschreibt die Einstellungen mit Zeilenumbrüchen.

Zurückgesetzt wird es mit:

```
:set nopaste
```

Was aber, wenn ich einen Zeilenumbruch habe und den gar nicht will oder der da gar nicht sein darf (s.o.)? Ich gehe in den Bewegungsmodus (ESC) und gehe auf die obere Zeile. Mit J (groß J) füge ich die untere Zeile mit der oberen zusammen.

Weitere wichtige Kommandos und Abläufe

Die Ein- und Ausgabekanäle

Kommandos geben oft regulären Output aus, aber auch Fehlermeldungen. Beim Starten eines Kommandos in der Shell, werden diese angezeigt. Oft ist es aber gewünscht, jeglichen Output umzulenken, z.B. wenn ein Kommando per cron (zeitgesteuert) und ohne Terminal (!) aufgerufen wird. Auch jegliche daemons kommen ohne Terminal aus, müssen aber auch mit ihren Ausgaben irgendwohin.

Ebenso kann der Output eines Kommandos als Input des nächsten Kommandos verwendet werden, oder eine Tastatureingabe dient als Input.

Wir unterscheiden also drei Kanäle:

0	Die Standardeingabe	STDIN
1	Der Standardoutput	STDOUT
2	Der Standarderror	STDERR

Ausgaben umleiten

Sagen wir, ich setze den Befehl ab:

```
ls bahnf*
```

Und erhalte als regulären Output:

```
bahnfahrten2004   bahnfahrten2007   bahnfahrten2010   bahnfahrten2013   bahnfahrten2016
bahnfahrten2005   bahnfahrten2008   bahnfahrten2011   bahnfahrten2014   bahnfahrten2017
bahnfahrten2006   bahnfahrten2009   bahnfahrten2012   bahnfahrten2015   bahnfahrten2018
```

Nun möchte ich das Ergebnis aber nicht in der Shell angezeigt haben, sondern, daß es in eine Datei landet. Ich lenke also die Ausgabe um:

```
ls bahnf* > allemeinebahnfahrten
```

Auf dem Terminal erscheint nun nichts mehr, nur der Prompt.

Allerdings können ja auch Fehler entstehen, gebe ich ein:

```
 ls bahnf* nase*
```

erhalte ich:

```
ls: cannot access 'nase*': No such file or directory
bahnfahrten2004   bahnfahrten2007   bahnfahrten2010   bahnfahrten2013   bahnfahrten2016
bahnfahrten2005   bahnfahrten2008   bahnfahrten2011   bahnfahrten2014   bahnfahrten2017
bahnfahrten2006   bahnfahrten2009   bahnfahrten2012   bahnfahrten2015   bahnfahrten2018
```

also zusätzlich eine Fehlermeldung.

Leite ich nun den Output um:

```
 ls bahnf* nase* > blubb
```

kommt nur noch auf dem Terminal:

```
 ls: cannot access 'nase*': No such file or directory
```

Hier sieht man sehr deutlich, daß Standardoutput und Standarderror verschiedene Kanäle sind.

Was aber, wenn ich den Error-Output auch nicht im Terminal brauchen kann? Ich leite erneut um:

```
 ls bahnf* nase* > blubb 2> bla
```

Das bedeutet: lenke den Standarderror in die Datei bla.

Möchte beide Kanäle in der gleichen Dateie haben, verwenden wir:

```
 ls bahnf* nase* > blubb 2>&1
```

Dies bedeutet: lenke den STDERR dahin, wo auch STDOUT hingeht.

> steht hier übrigens für "schreibe in Datei, wenn sie schon existiert, überschreibe den Inhalt".

>> steht für "hänge an eine bestehende Datei an, wenn sie nicht existiert, lege sie an".

Das Kommando

```
> bla
```

ist also eine einfache Art, Dateien zu leeren (z.B. Logfiles oder eben Ausgabefiles von Testkommandos etcpp.).

Andersherum kann man den Inhalt einer Datei als Eingabe eines Kommandos verwenden:

```
mail -s "Subject" foo@bar  < mail.txt
```

z.B. um eine oder mehrere Mails auf der Kommandozeile zu versenden.

Verkettung von Kommandos: die pipe

Der senkrechte Strich | wird als pipe bezeichnet. Er bedeutet:

"Nimm die Ausgabe von Kommando1 als Eingabe von Kommando2".

Dies kann z.B. nützlich sein, um aus dem Ergebnis von **ls -al** bestimmte Attribute zu filtern (z.B. mit grep s.u.) oder Spalten auszugeben (awk s.u.). Gern wird auch genommen: grep (nach Pattern in Zeilen suchen) und dann aber sortieren (sort), bzw. Dubletten aussortieren (sort -u).

In Vorgriff auf grep und Prozesse ein Beispiel:

```
ps -ef |grep princess
```

gibt alle Prozesse aus (ps -ef), gibt dann aber weiter und sucht nur nach dem pattern princess.

Weitere gute und nützliche Beispiele folgen! Auch und gerade hie ist es wichtig, eventuelle Fehlerausgaben NICHT in das nächste Kommando zu schicken!

Fileberechtigungen und wie man sie setzt (chmod)

Die lange Ausgabe von ls, ls -l gibt aus

- die Fileberechtigungen
- die Anzahl hardlinks
- Username

- Gruppenname
- Größe der Datei (mit -h in "human readable" statt in bytes ;-))
- Monat, Tag, Jahr oder Uhrzeit
- Filename

Die Fileberechtigungen zeigen an der ersten Stelle an, um was für eine Datei es sich handelt:

-	für eine normale Datei
d	für ein Directory
l	für einen symbolischen Link

(es gibt noch mehr, aber die meisten sieht man selten)

Sodann folgt je wein triplet aus

- Berechtigungen für den user
- Berechtigungen für die Gruppe
- Berechtigungen für alle anderen user auf dem Rechner

Bereits in den Rechten für das Home-Directory kann man viel falsch und richtigmachen. Das Home sollte nur für den user selbst lesbar sein. Gemeinsame Projekte sollten Gruppenrechte haben und in anderen Arbeitsverzeichnissen untergebracht sein (ist aber auch eine philosophische Frage).

Es gibt drei Rechte:

- r für lesen
- w für schreiben
- x für ausführen (execute). Bei Verzeichnissen heißt dieses Recht "hereinwechseln dürfen".

Beispiel:

```
drwx------
```

Heißt: in dieses Verzeichnis darf nur der Besitzer hereinwechseln und lesen und schreiben.

```
-rwxrwxrwx
```

heißt: diese Datei dürfen alle lesen, schreiben, ausführen.

```
-rwxr-xr-x
```

wird gern für ausführbare Kommandos genommen (**Aufgabe:** was kommt bei ls -l /bin/ls bei den Berechtigungen) und

```
-rw-r--r--
```

für Systemdateien (**Aufgabe:** ls -l /etc/passwd).

Geändert werden die Benachrichtigungen mit chmod.

```
chmod g+w datei
```

setzt Schreibrechte für die Gruppe. Man nennt also zuerst

- u für user, g für Group und o für others
- dann + für Recht hinzufügen und - für Recht entziehen
- und dann die Rechte r, w, x wie man es braucht.

Aufgabe: man kann statt den Buchstabenkombinationen auch Oktalzahlen verwenden, also 755 für rwxr-xr-x. Wie heißt die Zahl für rw-r--r--? (Ja das ist knifflig!)

Nutzer und Gruppe werden mit chown und chgroup geändert.

Aufgabe: lies beide manpages an.

Prozesse listen und killen

Wir haben schon erfahren, daß manchem Prozesse (daemons) ohne zugeordnetes Terminal auskommen.

Aber: alle Prozesse haben einen Elternprozess, von dem Eigenschaften "vererbt" werden und jeder Prozess hat eine eindeutige Nummer.

Der init-Prozess hat die 1 (und als Elternprozess die 0), alle weiteren Prozesse leiten sich von ihm ab.

Führt man in der Shell aus:

```
ps
```

so erhält man Auskunft, daß gerade eine bash läuft und ein ps:

```
  PID TTY          TIME CMD
49991 pts/0    00:00:00 bash
```

```
50183 pts/0    00:00:00 ps
```

Es wird die ProzessID ausgegeben (PID) das Terminal (tty), Zeit und das Kommando, also nur die gerade beteiligten Prozesse und Kindprozesse.

Es läuft auf einem Rechner natürlich viel mehr, diesen Output erhalten wir mit

```
ps -ef
```

und hier erhalten wir auch die PPID (Parent-Prozess-ID) und sehen, daß die meisten root-Prozesse bei TTY ein "?" stehenhaben und als PPIF 1 oder 2.

Aufgabe: leite den Output von ps -ef nach less und blättere seitenweise einmal durch. **Aufgabe 2:** leite den Output in eine Datei um und lies diese mit less.

Besonders deutlich wird das Eltern-Kind-Konzept, wenn man sich

```
 pstree
```

ansieht.

kill

Manchmal ist es notwendig, ein Kommando gewaltsam zu beenden, weil es nichts mehr tut oder amok läuft oder ohnehin irgendetwas unerwünschtes tut. IdR. kann man nur seine eigenen Prozesse beenden, root (sudo) kann alle Prozesse abschießen.

Üblicherweise reicht

```
kill PID
```

Wir haben ja schon gesehen, wie wir uns die benötigte Prozeßnummer beschaffen können. Man führt das Kommando einfach zweimal aus, um zu sehen, ob es geklappt hat. Meckert die Shell beim zweiten Mal NICHT, daß es den Prozess nicht mehr gibt, muß man zu harten Mitteln greifen:

```
kill -9 PID
```

Das "-9" bitte sparsam verwenden und nicht "einfach immer", denn es gibt dem Prozess keine Chance mehr, hinter sich aufzuräumen, also z.B. temporäre Dateien zu schließen und wieder zu entfernen.

Klappt auch das nicht, ist der Prozess ein Geist und erscheint in der Prozessliste als "defunct". Oft gehen diese Prozesse erst beim reboot wieder weg. Meist stören sie aber auch nicht.

Eine weitere wichtige Option ist -1

```
kill -1 PID
```

wird oft bei Systemprozessen eingesetzt und bedeutet: "Beende dich nicht, aber lies mal Deine Configfiles neu ein". Dies ist meist sparsamer, als den Systemprozess mit stop und start zu traktieren und geht auch oft schneller.

Hintergrundprozesse

Befehle, die wir in die Shell eintippen, "laufen im Vordergrund". Manche Prozesse laufen aber sehr lange, geben ihren Output eh nicht auf der Shell aus (s.o.) oder sind einfach graphische Prozesse, die ein neues Fenster aufmachen und die Shell nicht mehr brauchen.

Man möchte sie starten, braucht dann aber wieder einen Shellprompt.

Wenn man in einem Fenster ein neues Fenster starten will, kann man tippen:

```
xterm
```

(oder andere terminalfenster wie eterm, gnome-terminal, lxterm, ... Finde notfalls mit ps -ef heraus, wie Dein Terminal genau heißt).

Macht man aber das, kommt der Shellprompt nicht wieder. Er ist quasi von der laufenden Applikation xterm "belegt". Er kommt wieder, wenn Du das neue Fenster beendest, aber das ist ja nicht Sinn der Sache? ;-)

Man startet also grafische Programme "in den Hintergrund":

```
xterm &
```

Was aber, wenn ich das "&" vergessen habe?

Ich schicke den Prozess kurz "ins aus" mit **Ctrl-z** und sodann mit **bg** (background) in den Hintergrund und habe meine Shell wieder.

Mit **Ctrl-z** kann man auch andere Programme unterbrechen, z.B. das Abspielen von mp3s:

```
mpg321 lied.jpg
```

und wenn es weitergehen kann, holt man das Programm zurück mit **fg** für foreground.

Vererbung bei Prozessen

Die Bezeichnung Eltern- und Kindprozess kommt nicht von ganz ungefähr, denn es werden Eigenschaften der Elternprozesse an das Kind vererbt, z.B. Variablen.

Setzt man eine Variable z.B.

```
NASE="Nasenbaer"
```

so kann man sie, wie gelernt, mit "echo $NASE" ausgeben. Macht man nun ein xterm auf mit **xterm &** und versucht, die Variable dort auszugeben, gibt es sie nicht.

Erst wenn man die Variable **exportiert**, wird sie an den Kindprozess weitergegeben:

```
NASE="Nasenbaer"
export NASE
```

oder gleich:

```
export NASE="Nasenbaer"
```

Aufgabe: setze eine Variable und vererbe sie an ein Kind-xterm.

---

grep: Pattern in Dateien suchen

Eine der wichtigsten Tätigkeiten als Admine ist das Logfile-lesen. Geht etwas schief, schaut man erstmal ins log, was denn so loswar. Die meisten logs finden sich unter **/var/log**.

Zuerst schaut man sicher mit less herein und sucht ectl. Begriffe mit **/suchbegriff** wie im vi. (oder mit ?suchbegriff für "suche rückwärts").

Manchmal braucht man aber nur die Zeilen, in der ein bestimmtes pattern vorkommt, z.B. eine Mailadresse oder ein Loginname oder ein Teil einer URL. Dafür gibt es dann **grep**.

grep zeigt alle Zeilen an, die einen bestimmten Suchbegriff enthalten:

```
grep suchbegriff datei
```

Es kann sinnvoll sein, den Output nach less zu lenken und -i zu verwenden für "ignore case". Gerade bei Mailadressen wird gern Groß- und Kleinschreibung verwendet, auch wenn es keinen Unterschied macht:

```
grep -i suchbegriff datei | less
```

Weitere Varianten sind **egrep** (hier kann eine regular expression als suchbegriff verwendet werden), oder auch **rgrep**, das nicht in einer Datei sucht, sondern rekursiv in den unterliegenden Verzeichnissen.

zgrep kann in gezipten Files suchen (Endung .gz), ohne daß man erst die Datei auspacken muss (s.a.: zless).

Eine gute Anwendung für grep ist das Filtern des Output von ps -ef:

```
ps -ef | grep kommandoname
```

oder auch:

```
ps -ef | grep username
```

Man sieht hier aber auch: der grep-Prozess selber wird mit angezeigt. Wenn man eine Negation des Suchpatterns möchte, gibt es die Option **-v** (klein v):

```
ps -ef | grep kommandoname | grep -v grep
```

Eine sinnvolle Funktion für das .bashrc könnte also sein:

```
psg ()
{
   echo 'UID PID PPID C STIME TTY TIME CMD';
   ps -ef | grep $* | grep -v grep
}
```

Der Aufruf lautet dann also

```
psg suchbegriff
```

Aufgabe: warum muss die erste Zeile des Output von ps hier extra mit echo mit ausgegeben werden?

Aufgabe 2: schreibe die Funktion in Dein .bashrc und aktiviere sie.

awk: Spalten aus Dateien ausgeben

Eigentlich alle Konfigurationsdateien folgen einem strikten Aufbau und dies ist auch in den dazugehörigen manpages (Sektion 5) aufgeführt.

Aufgabe: lies man 5 crontab und man 5 passwd.

Oft sind die einzelnen Schlüsselworte durch Whitespaces getrennt (also Leerzeichen oder Tabulatoren) oder durch Trennzeichen wie der Doppelpunkt im passwd. Trennzeichen kennen wir auch von Exporten aus der Tabellenkalkulation (excel, librecalc), hier kann man das Trennzeichen beim Speichern selber wählen (z.B. Semikolon, Komma).

Zuweilen braucht man aber (für Skripte etcpp.) nur eine Spalte aus einer Datei oder einem Output. Dafür gibt es **awk**. Der Name setzt sich aus den Anfangsbuchstaben der Entwickler zusammen.

Beispiel: ich möchte anzeigen, welche Mailaliase definiert sind, aber nur die Namen, nicht, wo sie hingehen. Die Mailaliase befinden sich in **/etc/aliases**. Das Format der Datei ist in man 5 aliases beschrieben, das Trennzeichen ist also der Doppelpunkt.

```
awk -F: '{print $1}' /etc/aliases
```

Lasse ich die Angabe des Trennzeichens -F weg, wird white space als Trennzeichen angenommen. Die jetzige Ausgabe aber enthält noch die Kommentarzeilen, die mit # beginnen.

Aufgabe: filtere mit grep die Zeilen mit # heraus. Beachte, daß # ein Sonderzeichen ist.

Aufgabe 2: gib die erste Spalte von /etc/passwd aus.

Aufgabe 3: verbinde grep und awk und gib nur Deinen Usernamen aus /etc/passwd aus.

Ein weiteres Anwendungsbeispiel brauche ich beim "Archivieren von Hand": ls -al gibt neben Berechtigungen und Dateinamen auch das Datum aus, wann eine Datei zuletzt geschrieben wurde. Möchte ich z.B. alle Dateien aus dem Jahr 2017 archivieren/verschrieben, kann ich filtern:

```
ls -al |grep 2017
```

Die Dateien stehen im Output ganz hinten, ich kann nun mit der Maus cut'n'paste machen, aber son von Hand machen wir das natürlich nicht! :)

Ich möchte also nur den Dateinamen ausgeben, der steht (mit whitespace getrennt) in Spalte 9:

```
ls -al |grep 2017 |awk '{print $9}'
```

Sodann möchte ich ja, daß etwas mit den Dateien passiert, z.B. cp. Wie füttere ich aber dem cp nun diese Fileliste ein? Ich schließe den Befehl in Backquotes ein: ` (rückwärtiges Hochkomma). So wird in den Backquotes der Befehl durch das Ergebnis ersetzt:

```
cp `ls -al |grep 2017 |awk '{print $9}'` neuesverzeichnis
```

(neuesverzeichnis sollte vorher angelegt werden.

Nun ist die Syntax von awk, wie wir gesehen haben, etwas gewöhnungsbedürftig. Menschen in meinem U,fed deklarieren a1 bis a9 (oder sogar weiter) wie folgt:

```
alias a1="awk '{print \$1}'"
```

Wenn Du das sinnvoll findest, übernimm es in Dein .bashrc.

Wir sehen: mit grep, awk und der pipe stehen uns mächtige Werkzeuge zum Ansehen und Bearbeiten von Files und Output zur Verfügung.

Aufgabe: was macht im Gegensatz zu awk das Kommando **cut**?

---

find: Dateien nach bestimmten Kriterien anzeigen

find kann viel mehr als "nur" nach Namen suchen, sondern auch nach Permissions, Dateien oder Verzeichnissen, nach Alter einer Datei und ähnlichem. Desweiteren braucht man find auch, wenn ein lausiger Programmierer in einem Verzeichnis soviele Dateien angelegt hat, daß ls diese nicht mehr verarbeiten kann und aussteigt. Alles schon dagewesen!

find gibt unterhalb eines Verzeichnisses alle Dateien aus, die er rekursiv in allen Verzeichnissen findet:

```
find . -name '*rc' -print
```

Man gibt also zuerst das Verzeichnis an (. für das aktuelle Verzeichnis ist gern genommen), sodann die Suchkriterien. Verwendet man bei -name Wildcards, sind einfache Quotes ratsam. Der obige Befehl gibt also alle Files aus (auch die, die mit . beginnen!), die auf rc enden. Was war nochmal rc? :)

find ohne weitere Argumente gibt unterhalb von . alle Files aus, die er findet. -print braucht man unter Linux idR. nicht, aber es ist guter Stil, das sicherheitshalber immer mitzuschreiben.

Gut verbinden läßt sich find natürlich auch mit anderen Kommandos wie grep. Wenn ich also mal nicht weiß, in welcher Datei ich was abgelegt habe, kann ich suchen:

```
find . -type f -print | xargs grep suchbegriff
```

-type f sucht nur Dateien (in Verzeichnissen kann ich nich greppen, versuch es mal) und xargs brauchen wir hier aus diversen komplexen Gründen.

Aufgabe: lies Dich in den manpage ein um ein Gefühl zu bekommen, was find alles kann. Alternativ: suche im Ubuntuusers Wiki, dort ist auch vieles gut und mit Beispielen erklärt!