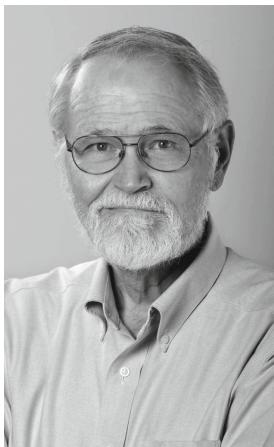


DIE UNIX STORY

Brian W. Kernighan

Die faszinierende Geschichte,
wie Unix begann und wie
es die Computerwelt eroberte

Brian Wilson Kernighan ist ein kanadischer Computerpionier (Jahrgang 1942). Er arbeitete lange Jahre in den Bell Labs und trug zusammen mit den Unix-Erfindern Ken Thompson und Dennis Ritchie zur Entwicklung von Unix bei.



Kernighans Name wurde durch die Co-Autorenschaft des ersten Buches über die Programmiersprache C (The C Programming Language) mit Dennis Ritchie weithin bekannt. Kernighan bekräftigte, dass er an der Gestaltung der Sprache C nicht beteiligt war (»es ist ausschließlich Dennis Ritchies Werk«).

Er ist Autor vieler Unix-Programme, einschließlich ditroff. Kernighan ist Mitautor der Programmiersprachen AWK und AMPL. Das »K« von K&R C und in AWK stehen beide für »Kernighan«. In Zusammenarbeit mit Shen Lin hat er bekannte Heuristiken für zwei NP-komplette Optimierungsprobleme entwickelt: die Graphenpartitionierung und das Problem des Handlungsreisenden.

Kernighan ist seit 2000 Professor für Informatik an der Universität Princeton und Leiter der »Undergraduate Studies« in der Informatikfakultät. Im Jahre 2015 war er Mitautor des Buches »The Go Programming Language«.

Brian W. Kernighan

Die UNIX-Story

Die faszinierende Geschichte, wie Unix begann
und wie es die Computerwelt eroberte



dpunkt.verlag

Brian W. Kernighan

Lektorat: Dr. Michael Barabas

Lektoratsassistenz: Anja Weimer

Übersetzung&Satz: G&U Language & Publishing Services GmbH, Flensburg,
www.GundU.com

Copy-Editing: Petra Heubach-Erdmann, Düsseldorf

Umschlaggestaltung: Helmut Kraus, www.exclam.de

Herstellung: Stefanie Weidner & Frank Heidt

Druckerei: mediaprint solutions GmbH, 33100 Paderborn

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

ISBN:

Print 978-3-86490-778-4

PDF 978-3-96910-072-1

ePub 978-3-96910-073-8

mobi 978-3-96910-074-5

1. Auflage 2021

Translation Copyright für die deutschsprachige Ausgabe © 2021 dpunkt.verlag GmbH
Wieblinger Weg 17
69123 Heidelberg

Brian W Kernighan: UNIX: A History and a Memoir, 1st Edition, ISBN 978-1695978553
© 2020 by Brian W. Kernighan. All rights reserved.

This work may not be translated or copied in whole or part without the written permission of the Author (www.kernighan.com).

Hinweis:

Dieses Buch wurde auf PEFC-zertifiziertem Papier aus nachhaltiger Waldwirtschaft gedruckt. Der Umwelt zuliebe verzichten wir zusätzlich auf die Einschweißfolie.



Schreiben Sie uns:

Falls Sie Anregungen, Wünsche und Kommentare haben, lassen Sie es uns wissen:
hallo@dpunkt.de.

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.

Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen. Alle Angaben und Programme in diesem Buch wurden mit größter Sorgfalt kontrolliert. Weder Autor noch Verlag noch Übersetzer können jedoch für Schäden haftbar gemacht werden, die in Zusammenhang mit der Verwendung dieses Buches stehen.

Inhalt

Vorwort	ix
Danksagung	xiii
Die Bell Labs	1
Naturwissenschaften in den Bell Labs	6
Kommunikationstechnik und Informatik	8
Meine Zeit in den Bell Labs	10
Die Büros	14
137 → 127 → 1127 → 11276	25
Das Ur-Unix (1969)	33
Technischer Hintergrund	33
CTSS und Multics	38
Die Ursprünge von Unix	40
Namen sind Schall und Rauch	43
Biografie: Ken Thompson	44
Erste Version (1971)	51
Unix für Patentanträge	53
Der Unix-Raum	57
Das Unix-Programmiererhandbuch	65
Anmerkungen zum Thema Speicher	67
Biografie: Dennis Ritchie	69
Sechste Version (1975)	75
Das Dateisystem	76
Systemaufrufe	78
Die Shell	81
Pipes	84
Grep	88
Reguläre Ausdrücke	93
Die Programmiersprache C	95
Ratfor und »Software Tools«	100
Biografie: Doug McIlroy	103

Siebte Version (1976–1979)	109
Die Bourne-Shell	110
Yacc, Lex und Make	113
Dokumenterstellung	123
Sed und Awk	141
Andere Sprachen	147
Weitere Errungenschaften	152
Jenseits der Forschung	163
Programmer's Workbench	164
Universitätslizenzen	168
Benutzergruppen und Usenix	170
John Lions' Kommentar	171
Portierbarkeit	174
Kommerzialisierung	177
Zerschlagung	177
USL und SVR4	179
UNIX™	181
Public Relations	183
Abkömmlinge	187
BSD (Berkeley Software Distribution)	189
Die Unix-Kriege	191
Minix und Linux	193
Plan 9	196
Diaspora	199
Das Vermächtnis	203
Technische Aspekte	204
Organisatorische Aspekte	210
Anerkennung	217
Könnte sich die Geschichte wiederholen?	220
Quellen	225
Quellen	225
Index	229

In memoriam DMR

Vorwort

»Eine der tröstlichen Eigenschaften von alten Erinnerungen ist, dass sie oft einen rosa Schimmer annehmen. Das Gedächtnis klammert sich an die Dinge, die gut waren und Bestand hatten, und an die Freude darüber, an Verbesserungen mitgewirkt zu haben, die das Leben erleichtern.«

– Dennis Ritchie, The Evolution of the Unix Time-sharing System, Oktober 1984

Seit das Betriebssystem Unix 1969 im Dachgeschoss der Bell Labs zur Welt gekommen ist, hat es sich weiter ausgebrettet, als seine Urheber es sich jemals hätten vorstellen können. Es hat zur Entwicklung einer großen Menge innovativer Software geführt, unzählige Programmierer beeinflusst und die Computertechnik in völlig neue Bahnen gelenkt.

Außerhalb eines bestimmten Kreises von Benutzern sind Unix und seine Abkömmlinge kaum bekannt, und doch bilden sie den Kern verschiedenster Systeme, die Bestandteil unseres Alltags geworden sind. Google, Facebook, Amazon und viele andere Dienste laufen auf Linux, einem Unix-artigen Betriebssystem, über das ich weiter hinten noch mehr schreiben werde. Auf Handys und Macs wird eine Unix-Version ausgeführt. Auch Geräte wie Alexa und Navigationssoftware für Autos nutzen Unix-artige Systeme. Wenn Sie im Web mit Werbung überschüttet werden, stecken ebenfalls Unix-Systeme dahinter, und die Trackingeinrichtungen, die herauszufinden versuchen, was Sie im Web machen, um Sie noch gezielter mit Reklame zu bombardieren, basieren sehr wahrscheinlich ebenfalls auf Unix.

Unix wurde vor über 50 Jahren hauptsächlich von zwei Personen sowie einer kleinen Gruppe von Mitarbeitern erschaffen. Aufgrund einer Kette glücklicher Zufälle war ich damals ebenfalls zugegen, wobei ich mir jedoch nichts davon als eigenen Verdienst anrechnen kann. Mein eigener Beitrag bestand höchstens aus einer bescheidenen Menge an nützlicher Software sowie – dank erstklassiger Co-Autoren – einigen Büchern, die anderen geholfen haben, mehr über Unix, seine Sprachen und seine Tools sowie über die dahinterstehende Philosophie zu lernen.

Dieses Buch ist zum Teil geschichtlicher Abriss und zum Teil persönliche Erinnerung. Es schildert die Ursprünge von Unix und versucht zu erklären, was Unix ist, wie es zustande kam und welche Bedeutung es hat. Ganz sicher ist es kein akademisches Werk – es enthält keine Fußnoten – und hat auch weniger von einem Geschichtswerk und dafür mehr von Memoiren bekommen, als ich ursprünglich geplant hatte.

Das Buch ist für alle gedacht, die sich für Computer oder Technikgeschichte interessieren. Es enthält auch einige technische Ausführungen, wobei ich aber stets versucht habe, ausreichende Erklärungen zu geben, damit auch Leser ohne umfassende Hintergrundkenntnisse die Grundprinzipien verstehen und deren Bedeutung erkennen können. Sie können jedoch auch problemlos über Stellen hinwegblättern, die Ihnen zu kompliziert erscheinen; es ist nicht nötig, jedes einzelne Wort zu lesen. Wenn Sie Programmierer sind, mögen Ihnen einige der Erklärungen überflüssig oder zu stark vereinfacht vorkommen, aber vielleicht finden die Ausführungen zur Geschichte von Unix und die damit verbundenen Anekdoten ja Ihr Interesse.

Ich habe mich zwar um Korrektheit bemüht, aber natürlich ist mein Gedächtnis nicht perfekt. Überdies lassen sich die Interviews, persönlichen Erinnerungen, Zeitzeugenberichte, Bücher und Artikel, auf die ich mich gestützt habe, nicht immer mit meinen Erinnerungen oder auch nur miteinander in Einklang bringen, wenn es darum geht, wer wann was gemacht hat.

Zum Glück sind viele derjenigen, die in der Frühzeit von Unix dabei waren, noch am Leben und konnten mich korrigieren. Auch sie haben natürlich mit Gedächtnislücken und der rosa Brille zu kämpfen, aber für jegliche im Text verbliebenen Fehler bin ich verantwortlich, zumindest sofern ich sie nicht mit Bestimmtheit jemand anderem anlasten kann.

Mein wichtigstes Anliegen beim Schreiben war es, einige der faszinierenden Geschichten aus einer besonders produktiven und prägenden Zeit in der Geschichte der Computer zu erzählen. Es ist wichtig, sich über die Entwicklung der Technologie im Klaren zu sein, die wir heute nutzen und für selbstverständlich halten. Die Entscheidungen, die diese Entwicklung bestimmten und damit unseren Weg vorgaben, wurden von Menschen getroffen, die dem Druck und den Einschränkungen ihrer Zeit ausgesetzt waren. Je mehr wir über die Geschichte wissen, umso mehr können wir das erforderliche Genie schätzen, das zu Unix geführt hat, und vielleicht auch besser verstehen, warum moderne Computersysteme so sind, wie sie sind. Zumindest kann es helfen, manche Entscheidungen, die aus heutiger Sicht falsch oder absurd erscheinen, als die natürlichen Folgen dessen zu erkennen, was man damals wusste und mit den verfügbaren Ressourcen erreichen konnte.

Es geht hier jedoch nicht nur um das Betriebssystem Unix, auch wenn es die Hauptsache darstellt, sondern auch um die Programmiersprache C. Sie gehört zu den am häufigsten verwendeten Sprachen und bildet das Herz der Systeme, die das Internet und die darin verfügbaren Dienste bereitstellen. Im Zusammenhang mit Unix kamen in den Bell Labs noch weitere Sprachen in die Welt, insbesondere das ebenfalls weitverbreitete C++, in dem Microsoft Office-Anwendungen wie Word, Excel und Power Point sowie die meisten Browser geschrieben sind. Ein oder zwei Dutzend der wichtigsten Werkzeuge, die Programmierer noch heute nutzen und für selbstverständlich halten, wurden in den Anfangstagen von Unix erstellt und sind nach 40 oder 50 Jahren zum Großteil noch unverändert.

Die theoretische Informatik spielte ebenfalls eine wichtige Rolle und ermöglichte oftmals die Entwicklung enorm praktischer Werkzeuge. Die Hardwareforschung untersuchte Designwerkzeuge, integrierte Schaltungen, Computerarchitekturen und ungewöhnliche Geräte für Sonderzwecke. Das Zusammenspiel all dieser Tätigkeitsbereiche führte oft zu unerwarteten Erfindungen und war einer der Gründe dafür, dass das gesamte Unternehmen auf so vielen verschiedenen Gebieten so produktiv war.

Es lassen sich auch einige bemerkenswerte und wichtige Einsichten darüber gewinnen, wie technischer Fortschritt zustande kommt. Die Bell Labs, in denen Unix seinen Anfang nahm, waren eine erstaunliche Einrichtung, die viele gute Ideen hervorbrachte und zu Geld machte. Hier

wurde eine Menge umwälzender Erfindungen gemacht, und aus der Art und Weise, wie das geschah, lässt sich viel lernen.

Die Geschichte von Unix vermittelt viele Erkenntnisse darüber, wie man Software entwirft und erstellt und wie man Computer wirkungsvoll einsetzt, was ich beim Schreiben auch hervorzuheben versucht habe. Ein simples, aber typisches Beispiel dafür ist die Unix-Philosophie der Software-Tools, nach der man einfach vorhandene Programme unterschiedlich kombiniert, um eine breite Palette von Aufgaben zu lösen, anstatt neue Software dafür zu schreiben. Durch die Aufteilung umfangreicher Aufgaben in kleinere lassen sich die einzelnen Teile nicht nur besser handhaben, sondern können auch auf zuvor unvermutete Weise kombiniert werden.

Unix war zwar die prominenteste Software aus den Bell Labs, aber auf keinen Fall deren einziger Beitrag zur Informationstechnologie. Das Computing Science Research Center – das legendäre »Center 1127« oder auch nur kurz »1127« – war zwei oder drei Jahrzehnte lang überaus produktiv. Es bezog Anregungen aus Unix und nutzte Unix als Grundlage für die Forschung, aber seine Errungenschaften gingen weit darüber hinaus. Die Mitglieder von 1127 schrieben bedeutende Bücher, die viele Jahre lang Standardwerke der Informatik und wichtige Nachschlagewerke für Programmierer darstellten. Das Center 1127 war ein außerordentlich einflussreiches kommerzielles Forschungslabor auf dem Gebiet der Informatik und eine der produktivsten Gruppen vergleichbarer Größe sowohl damals als auch in späterer Zeit.

Was hat Unix und sein Umfeld so erfolgreich gemacht? Wie konnte sich das Experiment zweier Personen zu etwas entwickeln, was die Welt veränderte? War dies ein einzigartiges und so unwahrscheinliches Ereignis, dass nie wieder mit etwas Ähnlichem zu rechnen ist? Zu der umfassenderen Frage, ob sich solche umwälzenden Ergebnisse planen lassen, werde ich mich gegen Ende des Buches noch ausführlicher äußern. Lassen wir es zunächst dabei bewenden, dass der Erfolg von Unix meiner Ansicht nach auf das zufällige Zusammentreffen verschiedener Faktoren zurückzuführen ist: zwei außergewöhnliche Menschen, hervorragende Mitarbeiter, ein fähiges und vorurteilsfreies Management, sichere Finanzierung durch ein Unternehmen mit einer langfristigen Sichtweise und eine Umgebung, die freie Forschung ungehindert erlaubte, wie unkonventionell sie auch immer sein mochte. Die Verbreitung von Unix wurde durch

den rasanten technischen Fortschritt gefördert, durch den die Hardware exponentiell kleiner, billiger und schneller wurde.

Die Anfangstage von Unix waren für mich und viele andere in den Bell Labs eine ausnehmend produktive und vergnügliche Zeit. Ich hoffe, Ihnen mit diesem Buch ein wenig von der Freude vermitteln zu können, etwas zu erschaffen und das Leben zu erleichtern, die Dennis Ritchie in dem einleitenden Zitat beschrieben hat.

Danksagung

Eine der unerwarteten Freuden beim Schreiben dieses Buches bestand darin, wieder mit so vielen alten Freunden und Kollegen in Berührung zu kommen. Sie waren so großzügig, mir ihre Erinnerungen mitzuteilen und einige wunderbare Geschichten zu erzählen. Wie wertvoll all dies für mich war, kann ich nur schwer in Worte fassen. Es war mir zwar nicht möglich, alle ihre Anekdoten hier zu verwenden, aber ich habe es sehr genossen, sie zu hören, und bin den vielen außergewöhnlichen Menschen zu Dank verpflichtet, mit denen ich zusammenarbeiten durfte.

Im ganzen Buch finden sich hier und da verstreut biografische Angaben zu verschiedenen Personen, wobei drei größere Abschnitte den drei Hauptverantwortlichen gewidmet sind, ohne die Unix nicht zustande gekommen wäre – Ken Thompson, Dennis Ritchie und Doug McIlroy. Ken und Doug haben mir außerordentlich wertvolle Rückmeldung zu dem Buch gegeben. Allerdings sind sie auf keinen Fall für irgendetwas verantwortlich zu machen, was ich falsch verstanden oder versehentlich falsch dargestellt haben sollte. Dennis' Brüder John und Bill haben ebenfalls nützliche Anmerkungen und Vorschläge gemacht, und sein Neffe Sam hat mehrere Entwurfsvorschläge ausführlich kommentiert.

Wie schon so oft zuvor hat mir Jon Bentley unschätzbar Einsichten vermittelt, hilfreiche Vorschläge zur Struktur des Buches und zur Schwerpunktsetzung gemacht, zahllose Anekdoten erzählt und mindestens ein halbes Dutzend Entwurfsvorschläge stilistisch kommentiert. Abermals bin ich Jon zu großem Dank verpflichtet.

Gerard Holzmann hat mir Ratschläge gegeben und Archivmaterial sowie viele Originalfotos zur Verfügung gestellt, die mir geholfen haben, das Buch aufzulockern.

Paul Kernighan hat mehrere Fassungen gelesen und dabei unzählige Tippfehler entdeckt. Außerdem hatte er einige hervorragende Vorschläge für den Titel, wobei ich mich jedoch schließlich widerstrebend gegen *A History of the Unix-speaking Peoples* (»Eine Geschichte der Unix-sprachigen Völker«) entschieden habe.

Al Aho, Mike Bianchi, Stu Feldman, Steve Johnson, Michael Lesk, John Linderman, John Mashey, Peter Neumann, Rob Pike, Howard Trickey und Peter Weinberger haben das Buch mit kritischem Auge gelesen und mir Geschichten aus den Anfangstagen von Unix erzählt, von denen ich viele zitiert oder in eigenen Worten wiedergegeben habe.

Viele hilfreiche Anmerkungen und sonstige Beiträge habe ich auch von Michael Bachand, David Brock, Grace Emlin, Maia Hamin, Bill Joy, Mark Kernighan, Meg Kernighan, William McGrath, Peter McIlroy, Arnold Robbins, Jonah Sinowitz, Bjarne Stroustrup, Warren Toomey und Janet Vertesi erhalten.

Ich bin sehr dankbar für die großzügige Unterstützung, aber für alle Irrtümer und Fehleinschätzungen bin ich allein verantwortlich. In den letzten fünfzig Jahren haben noch viele andere Menschen in der einen oder anderen Form wichtige Beiträge zu Unix geleistet, und ich möchte mich hier bei allen entschuldigen, deren Arbeit ich nicht erwähnt habe.

Die Bell Labs

»Eine Richtlinie, ein System, universeller Service.«

– Devise von AT&T, 1907

»In dieser unerwartet ländlichen Gegend von New Jersey wirkt der Hauptsitz der Bell Telephone Laboratories auf den ersten Blick wie eine große und moderne Fabrik, was er in gewissem Sinne auch ist. Allerdings handelt es sich um eine Ideenfabrik, weshalb die Fertigungsanlagen nicht zu sehen sind.«

– Arthur Clarke, *Voice Across the Sea*, 1974; zitiert nach *The Idea Factory* von Jon Gertner, 2012

Um nachvollziehen zu können, wie Unix in die Welt gekommen ist, müssen Sie zunächst die Bell Labs näher kennenlernen, insbesondere ihre Arbeitsweise und das kreative Milieu, das darin vorherrschte.

Das Unternehmen AT&T (American Telephone and Telegraph Company) entstand aus dem Zusammenschluss vieler örtlicher Telefonanbieter in den Vereinigten Staaten. Schon früh in der Firmengeschichte wurde klar, dass AT&T eine Forschungsgruppe benötigte, um systematisch all die wissenschaftlichen und technischen Probleme anzugehen, die sich dem Unternehmen bei der Bereitstellung landesweiter Telefondienste stellten. 1925 wurde dazu die Forschungsniederlassung Bell Telephone Laboratories gegründet. Auch wenn der Name gewöhnlich zu Bell Labs, BTL oder einfach nur Labs verkürzt wurde, kam der Telefontechnik immer zentrale Bedeutung zu.

Die Bell Labs waren ursprünglich in 463 West Street in New York City untergebracht, doch die meisten Tätigkeitsbereiche wurden zu Beginn



Abbildung 1.1

Der alte Standort in New York City und der neue in Murray Hill, New Jersey
(Karte: OpenStreetMap)



des Zweiten Weltkriegs nach außerhalb der Stadt verlegt. AT&T war ein kriegswichtiges Unternehmen und bot fachliche Beratung für ein breites Spektrum von militärischen Problemen – selbstverständlich für Kommunikationssysteme, aber auch für die Feuerleitrechner von Flugabwehr-geschützen, für Radar und Kryptografie. Ein Teil dieser Arbeit wurde in Vororten und ländlichen Gegenden von New Jersey erledigt. Der größte Standort befand sich in einem Gebiet namens Murray Hill, das zu den Kleinstädten New Providence und Berkeley Heights gehörte und 33 km westlich des alten Standorts lag.

Abbildung 1.1 zeigt die geografische Lage. 463 West Street befindet sich am Hudson River, auf der Karte ein wenig nördlich der Markierung für den Highway 9A. Die Bell Labs in Murray Hill liegen auf der Grenze zwischen New Providence und Berkeley Heights, etwas nördlich der Interstate 78. Beide Standorte sind mit roten Punkten markiert.

Immer mehr Tätigkeiten der Bell Labs wurden nach Murray Hill verlegt. Der Standort in der West Street wurde 1966 endgültig aufgegeben. In den 60er Jahren arbeiteten in Murray Hill mehr als 3.000 Personen, davon mindestens 1.000 mit einem Doktortitel in einer naturwissenschaftlich-technischen Disziplin wie Physik, Chemie, Mathematik oder verschiedenen Ingenieurwissenschaften.

Abbildung 1.2 zeigt eine Luftaufnahme der Einrichtungen in Murray Hill aus dem Jahr 1961. Aufgeteilt ist die Anlage in drei Hauptgebäude. Nr. 1 befindet sich in der Bildmitte und Nr. 2 oben links. Bei dem Vier-eck mit dem offenen Innenhof rechts handelt es sich um Nr. 3. Von dem einen Ende von Gebäude 1 bis zum anderen Ende von Gebäude 2 verlief damals ein durchgehender Korridor von 400 m Länge, der erst in den 70er Jahren durch zwei neu errichtete Gebäude unterbrochen wurde.

Ich habe mehr als 30 Jahre in Gebäude Nr. 2 gearbeitet, von meinem Praktikum im Jahr 1967 bis zu meiner Pensionierung 2000. Meine Büros befanden sich in den mit Punkten gekennzeichneten Seitenflügeln im vierten (obersten) Stock. Zur besseren Orientierung bei den folgenden Beschreibungen: Treppenhaus 9 befindet sich am hintersten Ende von Gebäude Nr. 2, Treppenhaus 8 einen Seitenflügel näher im Vordergrund. In den ersten Jahren war der Unix-Raum in der Dachkammer im fünften Stock zwischen diesen beiden Treppenhäusern untergebracht.

Abbildung 1.3 zeigt eine Google-Satellitenaufnahme der Bell Labs aus dem Jahr 2019. Die Gebäude Nr. 6 (unterhalb und links der Bildmitte, markiert) und Nr. 7 (oberhalb und rechts der Bildmitte) wurden Anfang der 70er hinzufügt. Nach 1996 fungierte Gebäude Nr. 6 einige Jahre lang als Hauptsitz von Lucent Technologies. Es ist faszinierend zu sehen, wie viel Firmengeschichte in den von Google hinzugefügten Beschriftungen steckt: Bell Labs als Hauptmarkierung, Lucent Bell Labs auf der Ausfahrt, Alcatel-Lucent Bell Labs an der Einfahrt und Nokia Bell Labs am Anbau der Verwaltungspyramide in Gebäude Nr. 6.

Ich bin nicht wirklich qualifiziert, um eine ausführliche Geschichte der Labs zu schreiben, aber das haben zum Glück schon andere Autoren erledigt. Mir persönlich gefallen besonders das Buch *The Idea Factory* von Jon Gertner, das den Schwerpunkt auf die physikalische Forschung legt, sowie *The Information* von James Gleick, eine hervorragende Quelle für die Tätigkeiten auf dem Gebiet der Informationswissenschaft. Die umfangreiche offizielle Publikation der Bell Labs (fast 5.000 Seiten in sieben Bänden) unter dem Titel *A History of Science and Engineering in the Bell System* ist ausführlich, zuverlässig und für meinen Geschmack äußerst interessant.

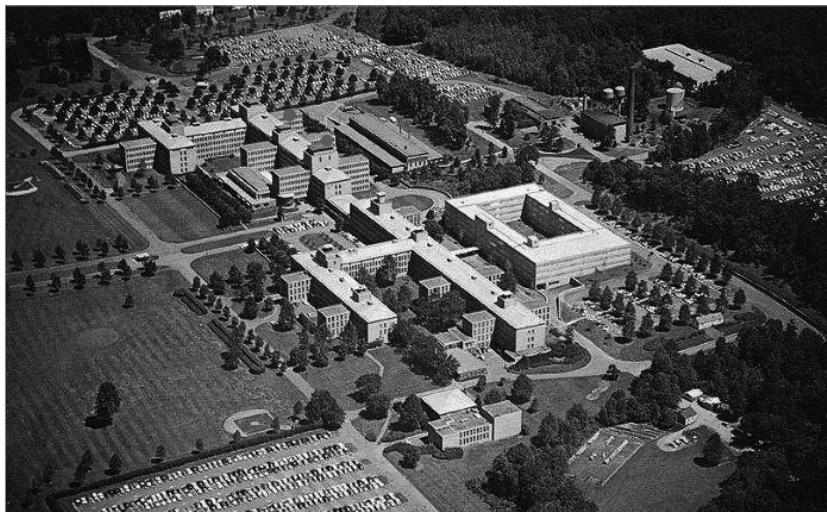


Abbildung 1.2

Die Bell Labs im Jahre 1961 (mit freundlicher Genehmigung der Bell Labs)

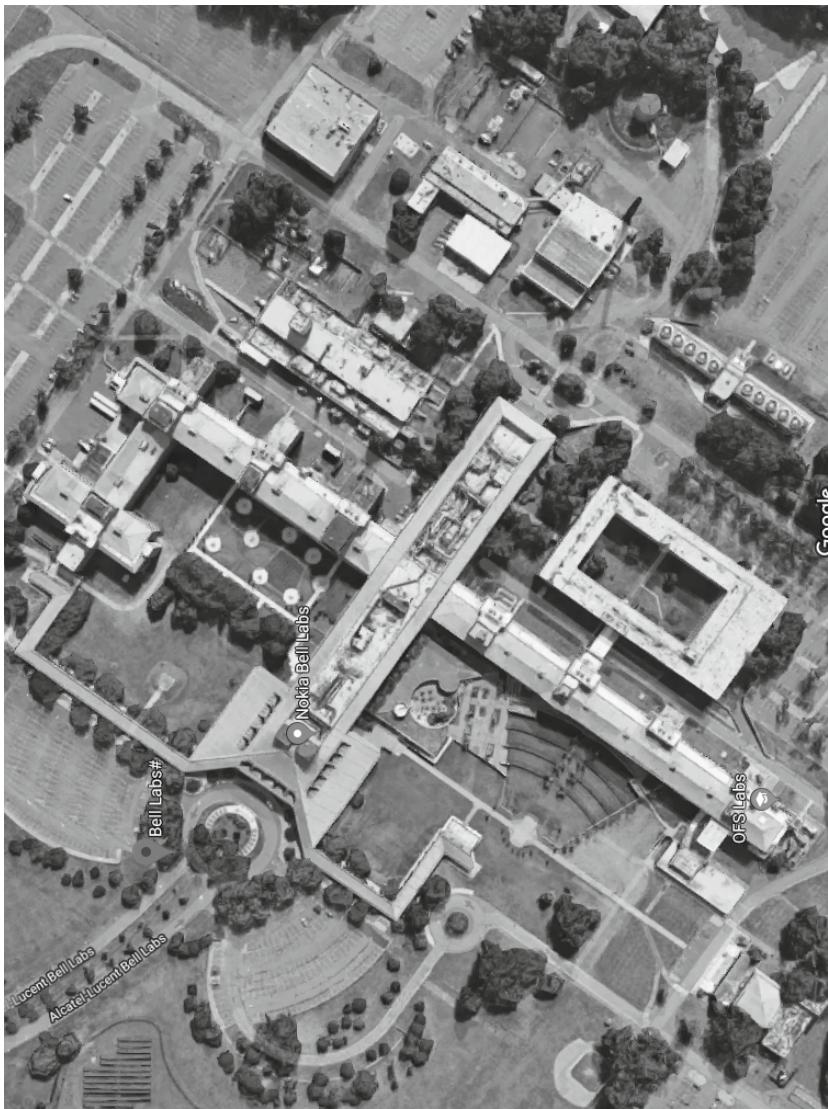


Abbildung 1.3

Die Bell Labs im Jahre 2019. Gebäude Nr. 6 liegt unten links.

Naturwissenschaften in den Bell Labs

In den ersten Jahren drehte sich die Forschung in den Bell Labs vorwiegend um Physik, Chemie, Materialwissenschaft und Kommunikationssysteme. Die Forscher genossen außerordentlich viel Freiheit, um ihren

eigenen Interessen nachzugehen, aber das Umfeld bot so viele relevante Probleme, dass es nicht weiter schwer war, sich auf Gebieten zu betätigen, die sowohl wissenschaftlich interessant waren als auch potenziellen Nutzen für Bell System und für die Welt im Allgemeinen versprachen.

Die Bell Labs steckten hinter einer großen Menge wissenschaftlicher und technischer Fortschritte von umwälzender Bedeutung. An erster Stelle steht dabei der Transistor, der 1947 von John Bardeen, Walter Brattain und William Shockley erfunden wurde, als sie daran arbeiteten, Verstärker für Ferntelefonleitungen zu verbessern. Der Transistor ging aus Grundlagenforschung über Halbleiter hervor, angeregt durch den Bedarf für Geräte, die haltbarer waren als Vakuumröhren und weniger Energie verbrauchten. Letztere bildeten in den 40er Jahren die einzige Möglichkeit, um Kommunikationsgeräte zu bauen – und nebenbei bemerkt auch die ersten Computer.

Die Erfindung des Transistors wurde mit dem Physik-Nobelpreis von 1956 gewürdigt – einem von neun Nobelpreisen für naturwissenschaftliche Arbeiten, die ganz oder teilweise in den Bell Labs erfolgten. Zu den weiteren bedeutenden Erfindungen gehören gegengekoppelte Verstärker, Solarzellen, Laser, Mobiltelefone, Kommunikationssatelliten und CCD-Sensoren (auf deren Grundlage Ihre Handykamera funktioniert).

Von den 60ern bis in die 80er arbeiteten etwa 3.000 Personen im Forschungsbereich der Bell Labs (vor allem in Murray Hill) und 15.000 bis 25.000 in Entwicklungsgruppen an verschiedenen Standorten, an denen Geräte und Systeme für die Bell Labs konstruiert wurden, oft auf der Grundlage von Ergebnissen des Forschungsbereichs. Das ist eine stattliche Anzahl. Wer hat sie alle bezahlt?

Da AT&T Telefondienste für fast die gesamten USA anbot, hatte das Unternehmen praktisch ein Monopol. Allerdings waren seine Möglichkeiten, diese Monopolstellung auszunutzen, beschränkt, da es von Behörden auf Bundes- und Staatenebene reguliert wurde. Die Preise, die AT&T für die verschiedenen Dienstleistungen verlangen durfte, unterlagen einer Kontrolle, und das Unternehmen durfte nur Geschäfte ausüben, die unmittelbar mit der Bereitstellung von Telefondiensten zusammenhingen.

Diese Regulierung funktionierte viele Jahre lang sehr gut. AT&T war verpflichtet, Dienstleistungen für alle anzubieten (»universeller Dienst«),

auch für abgelegene, unrentable Orte. Zum Ausgleich erhielt das Unternehmen eine stabile und vorhersehbare Rendite.

Im Rahmen dieser Vereinbarungen zweigte AT&T einen kleinen Teil seiner Einkünfte an die Bell Labs ausdrücklich für den Zweck ab, die Kommunikationsdienstleistungen zu verbessern. De facto wurden die Bell Labs also durch eine bescheidene Gebühr bezahlt, die bei jedem Telefonat im Land anfiel. Nach einer Abhandlung von A. Michael Noll gab AT&T etwa 2,8% der Einnahmen für Forschung und Entwicklung aus, wobei etwa 0,3% auf Grundlagenforschung entfielen. Ich bin mir nicht sicher, ob so etwas auch heute noch funktionieren würde, aber jahrzehntelang sorgte diese Vorgehensweise für kontinuierliche Verbesserungen des Telefonsystems und eine erhebliche Anzahl grundlegender wissenschaftlicher Entdeckungen.

Die sichere Finanzierung war von großer Bedeutung für die Forschung. Dadurch gewann AT&T eine sehr langfristige Sichtweise. Die Forscher bei den Bell Labs hatten die Freiheit, sich auch mit Themen zu beschäftigen, die keinen kurzfristigen Ertrag versprachen und vielleicht sogar niemals gewinnbringend sein mochten. Das unterscheidet sich sehr stark von der heutigen Situation, in der oft nur wenige Monate vorausgeplant und viel Aufwand betrieben wird, um über die finanziellen Ergebnisse im nächsten Quartal zu spekulieren.

Kommunikationstechnik und Informatik

Die Bell Labs leisteten naturgemäß Pionierarbeit bei der Konstruktion, Erstellung und Verbesserung von Kommunikationssystemen. Dieser Sammelbegriff umfasste alles – von der Endbenutzerhardware wie den Telefonen über Schaltsysteme und Mikrowellen-Sendeanlagen bis zu Glasfaserkabeln.

Manchmal führte diese breite Palette von praktischen Anwendungen sogar zu Fortschritten in der Grundlagenforschung. Beispielsweise versuchten Arno Penzias und Robert Wilson 1964 herauszufinden, was das störende Rauschen verursachte, das bei den von Echo-Ballonsatelliten zurückgeworfenen Funksignalen auftrat. Dabei erkannten sie schließlich, dass es sich um eine Hintergrundstrahlung handelte, die noch vom Urknall herrührte. Diese Entdeckung brachte ihnen 1978 den Nobelpreis ein. Penzias kommentierte dies mit den Worten: »Die meisten Leute

bekommen den Nobelpreis für Dinge, nach denen sie gesucht haben. Wir haben ihn für etwas bekommen, das wir loswerden wollten.«

Eine weitere Aufgabe der Bell Labs bestand darin, ein genaues mathematisches Verständnis der Funktionsweise von Kommunikationssystemen zu entwickeln. Das wichtigste Ergebnis dabei war die von Claude Shannon begründete Informationstheorie, die teilweise auf seinen Kryptografiestudien während des Zweiten Weltkriegs fußte. In seiner 1948 im *Bell System Technical Journal* veröffentlichten Abhandlung »A Mathematical Theory of Communication« beschrieb er die grundlegenden Eigenschaften von Information und fand heraus, wie man die Informationsmenge bestimmt, die prinzipiell über ein Kommunikationssystem übertragen werden kann. Shannon arbeitete von Anfang der 40er Jahre bis 1956 in Murray Hill. Danach kehrte er ans MIT zurück, wo er studiert hatte. 2001 starb er im Alter von 84 Jahren.

Als Computer immer leistungsfähiger und billiger wurden, erweiterten die Bell Labs ihren Einsatz nicht nur im Bereich der Datenanalyse, sondern dehnten ihn auch auf die umfassende Modellierung und Simulation von physikalischen Systemen und Prozessen aus. Die Bell Labs hatten sich schon seit den 30er Jahren mit Computern und EDV befasst und unterhielten seit Ende der 50er Jahre Rechenzentren mit zentralen Großrechnern.

Anfang der 60er wurden Mitarbeiter des Forschungszweiges Mathematik sowie Mitglieder des Bedienungspersonals für den großen Zentralcomputer in Murray Hill einer neu gegründeten Informatik-Forschungsgruppe zugewiesen, die die Bezeichnung Computing Science Research Center erhielt (nach der Nummer in der Organisationshierarchie oft auch »Center 1127« genannt). Auch wenn sie für kurze Zeit noch Computerdienste für alle Mitarbeiter in Murray Hill leistete, war sie keine Serviceeinrichtung, sondern gehörte zum Unternehmensbereich Forschung. 1970 wurde die Gruppe, die sich um die IT-Dienstleistungen kümmerte, in eine eigenständige Organisation ausgelagert.

Meine Zeit in den Bell Labs

Ein großer Teil dieses Abschnitts erzählt von meiner persönlichen Geschichte. Ich möchte Ihnen damit zeigen, welchen glücklichen Zufällen ich meine Karriere in der Informatik zu verdanken habe und dass die Bell Labs unschlagbar sind, wenn es darum geht, eine solche Karriere zu verfolgen.

Ich wurde in Toronto geboren und ging auch dort zur Universität. Mein Fachgebiet war Technische Physik, praktisch ein Sammelbecken für alle, die nicht genau wussten, auf welchen Schwerpunkt sie sich konzentrieren sollten. Meinen Abschluss machte ich 1964, in der Frühzeit der EDV. Ich war in meinem dritten Studienjahr, als ich zum ersten Mal einen Computer sah, nämlich den Großrechner, der für die gesamte Uni da war. Die IBM 7094 zählte damals zu den Spitzenmodellen und verfügte über einen Magnetkernspeicher für 32K (also 32.768) 36-Bit-Wörter (heute würden wir sagen 128 KByte) sowie Sekundärspeicher in Form großer, mechanischer Plattenlaufwerke. Sie kostete drei Millionen Dollar und war in einem großen Raum mit Klimaanlage untergebracht, behütet von professionellem Bedienpersonal. Gewöhnliche Sterbliche (insbesondere Studenten) wurden nicht einmal in die Nähe gelassen.

Daher habe ich mich während meines Diplomstudiums nur wenig mit Informatik beschäftigt. Allerdings lernte ich die Programmiersprache Fortran. Deshalb kann ich mich nur zu gut in Menschen hineinversetzen, die Schwierigkeiten haben, ihr erstes Programm zu schreiben. Ich hatte zwar Daniel McCrackens hervorragendes Buch über Fortran II gelesen und alle Regeln parat, aber mir war nicht klar, wie ich überhaupt beginnen sollte, um ein Programm zu schreiben. Dieser erste Schritt scheint vielen Einsteigern Schwierigkeiten zu bereiten.

Im Sommer vor meinem letzten Collegejahr hatte ich einen Job bei Imperial Oil in Toronto als Mitglied einer Gruppe, die Optimierungssoftware für Raffinerien entwickelte. (Imperial Oil gehörte zum Teil der Firma Standard Oil of New Jersey, aus der 1972 Exxon wurde.)

Im Nachhinein muss ich zugeben, dass ich wohl ein ziemlich unterdurchschnittlicher Praktikant war. Ich verwendete den ganzen Sommer darauf, ein riesiges Cobol-Programm zur Analyse von Raffineriedaten zu schreiben. An den genauen Zweck kann ich mich nicht mehr erinnern, aber ich weiß noch genau, dass es niemals funktionierte. Mir war nicht klar, wie

man überhaupt programmiert, Cobol bot wenig Unterstützung für eine gute Programmierung, und strukturierte Programmierung war noch nicht erfunden. Mein Code bestand aus einer endlosen Folge von IF-Anweisungen mit Verzweigungen, bei denen der Code irgendetwas tun sollte, sobald ich herausgefunden hatte, was dieses Etwas war.

Ich versuchte auch, einige Fortran-Programme auf der IBM 7010 von Imperial ans Laufen zu bekommen. Schließlich hatte ich gewisse Kenntnisse in Fortran – auf jeden Fall bessere als in Cobol –, und Fortran hätte sich auch besser für die Datenanalyse geeignet. Erst nachdem ich mich wochenlang mit JCL herumgeschlagen hatte, der »Job Control Language« von IBM, stellte ich fest, dass der 7010 keinen Fortran-Compiler hatte. Die Fehlermeldungen von JCL waren so undurchschaubar, dass es bis dahin auch niemand sonst herausgefunden hatte.

Als ich nach diesem etwas frustrierenden Sommer für mein Abschlussjahr an die Universität zurückkehrte, war ich immer noch sehr stark an Computern interessiert. Es gab keine formalen Informatikkurse, aber dennoch schrieb ich meine Diplomarbeit über künstliche Intelligenz, was damals ein gefragtes Thema war. Sogenannte »Theorembeweiser« oder Programme, die Schach und Dame spielen können, sowie die maschinelle Übersetzung menschlicher Sprachen schienen damals alle in Reichweite zu sein und nur noch ein bisschen Programmierung zu erfordern.

Nachdem ich 1964 meinen Abschluss gemacht hatte, wusste ich nicht recht, was ich als Nächstes tun sollte. Wie viele andere Studenten verschob ich die Entscheidung, indem ich mich zu einem Aufbaustudium entschloss. Ich bewarb mich bei einem Dutzend Universitäten in den USA (was damals unter Kanadiern nicht üblich war – und ich bin aus Kanada) und erhielt von mehreren eine Zusage, darunter vom MIT und von Princeton. Princeton teilte mir mit, dass normalerweise drei Jahre bis zur Promotion vergingen, während es laut Aussage des MIT wahrscheinlich sieben Jahre sein würden. Außerdem bot mir Princeton ein komplettes Stipendium an, während ich beim MIT 30 Stunden die Woche als Forschungsassistent zu arbeiten hätte. Das machte die Entscheidung ziemlich einfach. Außerdem studierte Al Aho, ein guter Freund, der in Toronto ein Jahr über mir gewesen war, bereits in Princeton. Also machte ich mich auf den Weg, was sich später als eine äußerst vorteilhafte Entscheidung herausstellte.

1966 hatte ich erneut Glück, da ich ein Sommerpraktikum beim MIT ergattern konnte. Teilweise lag das daran, dass Lee Varian, ebenfalls Princeton-Student, dort im Vorjahr hervorragende Arbeit geleistet hatte. Den Sommer über nutzte ich das CTSS (Compatible Time-Sharing System) und schrieb Programme in MAD (Michigan Algorithm Decoder, ein Dialekt von Algol 58), um Tools für das neue Betriebssystem Multics zu schreiben, auf das ich in Kapitel 2 noch zurückkommen werde. (Multics wurde ursprünglich MULTICS geschrieben, aber die Variante mit Kleinbuchstaben wirkt optisch angenehmer. Ich verwende hier die besser lesbare Version, auch wenn sie historisch nicht korrekt ist. Das gilt auch für andere komplett in Großbuchstaben geschriebene Wörter, also z.B. UNIX/Unix.)

Mein offizieller Vorgesetzter beim MIT war Prof. Fernando Corbató, von allen nur »Corby« genannt, ein wunderbarer Gentleman, der nicht nur CTSS entworfen hatte, sondern auch für Multics verantwortlich war. Für seine grundlegenden Arbeiten an Timesharing-Systemen erhielt er den Turing Award 1990. Corby starb im Juli 2019 im Alter von 93 Jahren.

Corby leitete nicht nur das Design und die Implementierung von CTSS und Multics, sondern war auch der Erfinder von Passwörtern für den Computerzugriff. Im Stapelverarbeitungsbetrieb war so etwas nicht nötig gewesen, aber bei einem Mehrbenutzercomputer mit einem gemeinsamen Dateisystem war ein Sicherheitsmechanismus zum Schutz der privaten Dateien erforderlich.

Ich kann mich noch an die beiden kurzen, aussprechbaren Passwörter erinnern, die für mich im Sommer 1966 am MIT generiert wurden. Es waren zwei, nicht nur eines, da eines Tages jemand die Datei mit den Passwörtern und die Datei mit der Login-Meldung des Tages verwechselte. Da die Passwörter unverschlüsselt gespeichert waren, konnte dadurch jeder, der sich anmeldete, sämtliche Passwörter einsehen.

Mein Büro gehörte eigentlich Prof. Joseph Weizenbaum, der den Sommer über aber nicht da war, sodass ich es nicht mit ihm teilen musste. Anfang 1966 hatte Weizenbaum eine Beschreibung von Eliza veröffentlicht, einem Programm, das ein Gespräch mit einem Psychologen der Rogers-Schule simulierte. Eliza stellte einen der ersten Versuche dar, zur Interaktion mit einem Computer auf natürliche Sprache zurückzugreifen. Die Simulation wirkte erstaunlich überzeugend, zumindest wenn man