Lernhilfe: Unix-Zeitreise

Fünf historische Unix-Systeme von 1969 bis 1986 im Emulator Open SIMH • Docker

Lernziele

- Fünf historische Unix-Systeme im Emulator starten und bedienen
- Erkennen, welche Befehle und Konzepte von Linux schon zwischen 1969 und 1986 entstanden sind
- Den Weg vom Assembler über die Sprache B zu C nachvollziehen
- Ein Unix von 1986 per TCP/IP ins Internet pingen lassen
- Mehrere Systeme mit Docker Compose betreiben und ihre CPU-Last begrenzen

Voraussetzungen

- Docker mit Compose, zum Beispiel in der Debian-VM oder mit Docker Desktop
- Der Ordner **Unix-Zeitreise** mit Dockerfile, compose.yaml und start.sh
- Internetzugang für den ersten Build und für zwei fertige Images aus Docker Hub
- Optional ein Telnet-Client (telnet oder PuTTY) für weitere Sitzungen

Aufbau

Die Lernhilfe führt in fünf Stationen durch die Geschichte von Unix, jede Station ist ein System. Alle Stationen sind gleich aufgebaut:

- **Hintergrund:** was das System besonders macht
- **Starten:** Docker-Befehl und Login
- **Aufgaben:** Befehle zum Abtippen und was Sie dabei sehen sollten
- **Mini-Übung:** ein kleines Programm oder Experiment

Der Zeitstrahl am rechten Rand zeigt, wo Sie sich gerade in der Geschichte befinden.

Vorbemerkung: Was ist ein Emulator?

Die Originalrechner von DEC (PDP-7, PDP-11, VAX) stehen heute fast nur noch im Museum. Ein Emulator bildet Prozessor, Speicher, Platten und Terminals eines solchen Rechners in Software nach. Darauf laufen die originalen Betriebssysteme unverändert. Hier kommt der Emulator **Open SIMH** zum Einsatz, fertig verpackt in Docker-Images.

Station	System	Jahr	Rechner	Login
1	PDP-7-Unix	1969/70	DEC PDP-7	ken / ken
2	Unix V1	1971/72	DEC PDP-11/20	root
3	Unix V4	1973	DEC PDP-11/45	root
4	Unix V7	1979	DEC PDP-11/45	root / root
5	4.3BSD	1986	DEC VAX-11/780	root

Vorbereitung (einmalig)

Im Ordner Unix-Zeitreise:

```
docker compose up -d
```

Beim ersten Mal dauert das etwa zwei Minuten: Docker übersetzt den Emulator, baut PDP-7-Unix und Unix V1 aus den restaurierten Quelltexten und lädt die Images für V7 und 4.3BSD. Danach laufen alle fünf Systeme im Hintergrund als Container pdp7, v1, v4, v7 und bsd43.

Zeitstrahl

Unix von 1969 bis heute

- **1969** — PDP-7-Unix
- **1971** — Unix V1
- **1973** — Unix V4: Kernel in C
- **1975** — Unix V6 an Hochschulen
- **1978** — 1BSD aus Berkeley
- **1979** — Unix V7
- **1983** — 4.2BSD: TCP/IP, Sockets
- **1986** — 4.3BSD
- **1988** — POSIX-Standard
- **1991** — Linux 0.01
- **1993** — FreeBSD, NetBSD
- **2001** — Mac OS X (BSD-Wurzeln)
- **2025** — V4-Band in Utah gefunden
- **heute** — Linux, macOS, Android, BSD

1 Station in dieser Lernhilfe

Zwei Wege zum System

	Compose: alle im Hintergrund	docker run: einzeln
Starten	`docker compose up -d`	Befehl aus der jeweiligen Station
Konsole öffnen	`docker attach v1`, ebenso pdp7, v4, v7 und bsd43. Der Bildschirm ist zuerst leer: einfach den Login tippen.	Die Konsole erscheint direkt im Terminal.
Verlassen	**Strg+P**, dann **Strg+Q**. Das System läuft weiter.	**Strg+E**, dann `quit`. Das System ist beendet.
Zustand	Bleibt erhalten, bis Sie `docker compose down` eingeben.	Mit `--rm` startet jedes Mal ein frisches System.
Anhalten	`docker compose stop`, wieder starten mit `docker compose start`	entfällt

Bedienung: Das gilt an allen Stationen

Situation	So geht es
Emulator beenden	**Strg+E** hält den Emulator an, danach `quit` eingeben. Bei V7 und 4.3BSD beendet Strg+E den Container sofort.
Tippfehler (Station 1 bis 4)	Die Rücktaste funktioniert hier nicht wie gewohnt. `#` löscht das letzte Zeichen, `@` die ganze Zeile.
Tippfehler (Station 5)	Die Rücktaste funktioniert, **Strg+U** löscht die ganze Zeile.
Datum und Uhrzeit	Die Uhren der Systeme sind nicht gestellt. Ein seltsames Datum ist also kein Fehler.
CPU-Last	Die Emulatoren schlafen im Leerlauf oder sind gedrosselt. Compose begrenzt jeden Container zusätzlich auf einen halben CPU-Kern, bei docker run geht das mit `--cpus=0.5`.

Zeitstrahl

Unix von 1969 bis heute

- **1969** (1) PDP-7-Unix
- **1971** (2) Unix V1
- **1973** (3) Unix V4: Kernel in C
- **1975** Unix V6 an Hochschulen
- **1978** 1BSD aus Berkeley
- **1979** (4) Unix V7
- **1983** 4.2BSD: TCP/IP, Sockets
- **1986** (5) 4.3BSD
- **1988** POSIX-Standard
- **1991** Linux 0.01
- **1993** FreeBSD, NetBSD
- **2001** Mac OS X (BSD-Wurzeln)
- **2025** V4-Band in Utah gefunden
- **heute** Linux, macOS, Android, BSD

(1) Station in dieser Lernhilfe

Station 1: PDP-7-Unix (1969/70)

Im Sommer 1969 schrieb Ken Thompson bei den Bell Labs auf einem wenig genutzten DEC PDP-7 das erste Unix, komplett in Assembler. Der Rechner hatte 8192 Speicherwörter zu je 18 Bit. Bald arbeitete Dennis Ritchie mit, die beiden gelten als Erfinder von Unix. Der Quelltext überlebte nur als Ausdruck. 2016 haben Freiwillige ihn abgetippt und das System wieder zum Laufen gebracht.

Starten:

```
docker run -it --rm unix-zeitreise pdp7
```

Mit Compose: `docker attach pdp7`

Login: `ken`, Passwort `ken`. Der Prompt ist ein `@`.

Aufgaben

Befehl	Was Sie sehen
`ls`	Ihr Home-Verzeichnis. Die Dateien s1.s bis s8.s sind der Quelltext des Kernels in PDP-7-Assembler.
`cat s1.s`	Der Anfang des Kernels. Kommentare beginnen mit einem Anführungszeichen (").
`ls system`	Die Programme des Systems, darunter cat, cp, ed, ls, mv und sh (die Shell).
`date`	`Thu Jan 01 1970` und ein paar Sekunden: Die Uhr ist nicht gestellt und zählt ab dem Start.
`cd dd`	Die Shell kennt keinen Befehl cd und antwortet nur mit `cd ?`. Fehlermeldungen waren damals sehr knapp.

Mini-Übung: Hello, World! in B

B ist die Sprache, aus der später C wurde. Beenden Sie den Emulator (Strg+E, `quit`), starten Sie ihn neu und melden Sie sich als **dmr** mit Passwort **dmr** an. Im Home-Verzeichnis von Dennis Ritchie liegt ein kleines B-Programm:

```
cat hello.b
b hello.b hello.s
as op.s bl.s hello.s bi.s
a.out
```

Erwartete Ausgabe:

```
Hello, World!
```

b übersetzt das B-Programm in Assembler, **as** fügt es mit den Bausteinen der B-Laufzeitumgebung zu **a.out** zusammen. Diesen Namen für ein frisch übersetztes Programm gibt es bis heute. In hello.b steht `write('He')`: Ein Speicherwort fasst genau zwei Zeichen, `*n` ist der Zeilenumbruch.

> **Info: Woher kommt der Name Unix?**
>
> Die Bell Labs hatten vorher am riesigen Mehrbenutzersystem Multics mitgearbeitet. Das neue, kleine System bekam als Wortspiel darauf den Namen Unics, daraus wurde Unix.

Zeitstrahl

Unix von 1969 bis heute

- **1969** — PDP-7-Unix
- **1971** — Unix V1
- **1973** — Unix V4: Kernel in C
- **1975** — Unix V6 an Hochschulen
- **1978** — 1BSD aus Berkeley
- **1979** — Unix V7
- **1983** — 4.2BSD: TCP/IP, Sockets
- **1986** — 4.3BSD
- **1988** — POSIX-Standard
- **1991** — Linux 0.01
- **1993** — FreeBSD, NetBSD
- **2001** — Mac OS X (BSD-Wurzeln)
- **2025** — V4-Band in Utah gefunden
- **heute** — Linux, macOS, Android, BSD

1 Station in dieser Lernhilfe

Station 2: Unix V1 (1971/72)

Im November 1971 erschien die erste offizielle Ausgabe, die First Edition. Unix lief jetzt auf dem neuen DEC PDP-11/20, der Kernel war weiterhin in Assembler geschrieben. Auch dieser Quelltext blieb nur als Ausdruck von 1972 erhalten und wurde 2008 abgetippt.

Starten:

```
docker run -it --rm unix-zeitreise v1
```

Mit Compose: `docker attach v1`

Login: `root`, ohne Passwort. Der Prompt ist `#`.

Aufgaben

Befehl	Was Sie sehen
`cd /usr`	`No command`: Den Befehl cd gibt es noch nicht.
`chdir /usr`, dann `ls`	So wechselte man 1971 das Verzeichnis. In /usr liegen unter anderem ken und sys.
`ls -l`	Die Rechte sehen anders aus: `sdrwr-` heißt kleine Datei (s), Verzeichnis (d), der Besitzer liest und schreibt (rw), alle anderen lesen nur (r-).
`cat /etc/passwd`	Die Benutzerliste, darunter ken (Ken Thompson) und dmr (Dennis Ritchie). Das Feld für das Passwort ist leer.
`date`	Datum und Uhrzeit, aber ohne Jahreszahl.
`who`	Wer gerade angemeldet ist, damals wie heute.
`ls /bin`	Viele Programme kennen Sie: cat, cp, ed, ls, mkdir, mv, rm, sort, wc und who.

Mini-Übung: Mehrere Benutzer gleichzeitig

Unix war von Anfang an ein Mehrbenutzersystem. Starten Sie V1 mit freigegebenem Port (bei Compose ist der Port schon freigegeben):

```
docker run -it --rm -p 5555:5555 unix-zeitreise v1
```

Öffnen Sie ein zweites Terminal und verbinden Sie sich per Telnet (unter Windows zum Beispiel mit PuTTY, Verbindungstyp Telnet, unter Debian nach `sudo apt install telnet`):

```
telnet localhost 5555
```

Melden Sie sich in beiden Sitzungen als root an und geben Sie `who` ein. Tippen Sie langsam: Zu schnell eingefügter Text kommt bei der alten Terminalschnittstelle nicht vollständig an.

> **Info: Die Unix-Philosophie**
>
> Viele kleine Programme, die jeweils eine Sache gut erledigen und sich miteinander kombinieren lassen. Diese Idee prägt Unix seit den 1970er-Jahren und gilt bis heute für Linux.

Zeitstrahl

Unix von 1969 bis heute

- **1969** — PDP-7-Unix
- **1971** — Unix V1 *(aktuell)*
- **1973** — Unix V4: Kernel in C
- **1975** — Unix V6 an Hochschulen
- **1978** — 1BSD aus Berkeley
- **1979** — Unix V7
- **1983** — 4.2BSD: TCP/IP, Sockets
- **1986** — 4.3BSD
- **1988** — POSIX-Standard
- **1991** — Linux 0.01
- **1993** — FreeBSD, NetBSD
- **2001** — Mac OS X (BSD-Wurzeln)
- **2025** — V4-Band in Utah gefunden
- **heute** — Linux, macOS, Android, BSD

1 Station in dieser Lernhilfe

Station 3: Unix V4 (1973)

Dennis Ritchie entwickelte aus B die Sprache C. 1973 schrieben Thompson und Ritchie den Kernel in C neu: Unix V4 vom November 1973 ist die erste Version mit einem Kernel in einer Hochsprache. Dadurch ließ sich Unix später auf andere Rechner übertragen.

Starten:

```
docker run -it --rm unix-zeitreise v4
```

Mit Compose: `docker attach v4`

Nach dem Start erscheint keine Eingabeaufforderung. Tippen Sie **k** (ohne Enter) und danach **unix** mit Enter. Der Kernel meldet sich mit `mem = 64530`.

Login: `root`, ohne Passwort.

Aufgaben

Befehl	Was Sie sehen
`cd /usr`	`cd: not found`: Auch 1973 heißt der Befehl noch chdir.
`chdir /usr/sys`, dann `ls`	Header-Dateien wie proc.h und user.h: Der Kernel ist in C geschrieben.
`ls ken`	Die Kernel-Quellen von Ken Thompson, zum Beispiel main.c, clock.c und trap.c.
`cat ken/main.c`	C-Code von 1973 mit Copyright der Bell Labs (dafür nach oben scrollen). Statt += steht dort noch =+. Die Zeile mit `printf("mem = %l\n", ...)` erzeugt genau die Meldung beim Booten.
`ls -l /`	Die Rechte sehen jetzt aus wie heute, zum Beispiel `drwxr-xr-x`.
`date`	Ein Datum aus dem Jahr 1974.

Mini-Übung: Hallo Welt in C von 1973

```
chdir /tmp
cat >hallo.c
main() {
    printf("Hallo aus dem Jahr 1973\n");
}
```

Nach der letzten Zeile beendet **Strg+D** die Eingabe. Dann übersetzen und starten:

```
cc hallo.c
a.out
```

Erwartete Ausgabe:

```
Hallo aus dem Jahr 1973
```

So sah C damals aus: ohne `#include` und ohne `int` vor main. Ein `#` dürfen Sie hier ohnehin nicht tippen, es löscht ja das vorige Zeichen.

> **Info: Ein Band aus dem Abstellraum**
>
> Von Unix V4 gab es lange keine vollständige Kopie. 2025 tauchte an der University of Utah ein Magnetband von 1974 auf. Das Computer History Museum hat es ausgelesen, Angelo Papenhoff hat daraus das Plattenabbild gebaut, das Sie hier starten.

Zeitstrahl

Unix von 1969 bis heute

1. **1969** PDP-7-Unix
2. **1971** Unix V1
3. **1973** Unix V4: Kernel in C
4. **1975** Unix V6 an Hochschulen
5. **1978** 1BSD aus Berkeley
6. **1979** Unix V7
7. **1983** 4.2BSD: TCP/IP, Sockets
8. **1986** 4.3BSD
9. **1988** POSIX-Standard
10. **1991** Linux 0.01
11. **1993** FreeBSD, NetBSD
12. **2001** Mac OS X (BSD-Wurzeln)
13. **2025** V4-Band in Utah gefunden
14. **heute** Linux, macOS, Android, BSD

1 Station in dieser Lernhilfe

Station 4: Unix V7 (1979)

Die Version 7 vom Januar 1979 gilt als Stammvater fast aller späteren Unix-Systeme. Neu waren unter anderem die Bourne-Shell mit Variablen und Schleifen, Umgebungsvariablen sowie Programme wie awk und tar. Dieses Image stammt von Jordi Guillaumes und liegt fertig auf Docker Hub.

Starten:

```
docker run -it --rm -e SIMH_USE_CONTAINER=yes jguillaumes/simh-pdpv7
```

Mit Compose: `docker attach v7`

Booten von Hand: Tippen Sie `boot`, beim Doppelpunkt `hp(0,0)unix` und beim ersten # **Strg+D**. Erst dann startet der Mehrbenutzerbetrieb mit Login.

Login: `root`, Passwort `root`. Die Konsole läuft absichtlich mit 300 Baud, die Zeichen erscheinen so langsam wie auf einem Terminal von 1979.

Aufgaben

Befehl	Was Sie sehen
`cd /usr`, dann `pwd`	Endlich gibt es cd. pwd zeigt `/usr`.
`echo $HOME $PATH`	Umgebungsvariablen, neu in V7: `/` und `:/bin:/usr/bin`.
`ps`	Die Prozessliste mit PID, Terminal und Laufzeit, fast wie heute.
`ls /bin | wc -l`	Eine Pipe verbindet zwei Programme (Pipes gibt es seit 1973). wc zählt 154 Programme.
`cat /etc/passwd`	Im zweiten Feld steht jetzt ein Passwort-Hash. Heute liegt er in /etc/shadow.

Mini-Übung: Ein Shell-Skript von 1979

```
cd /tmp
cat >hallo.sh
for name in Ken Dennis Steve
do
    echo Hallo $name
done
```

Strg+D beendet die Eingabe. Dann starten Sie das Skript:

```
sh hallo.sh
```

Erwartete Ausgabe:

```
Hallo Ken
Hallo Dennis
Hallo Steve
```

Steve Bourne hat diese Shell geschrieben. Die Schleife funktioniert unverändert in der Bash.

> **Info: Warum V7 so wichtig ist**
>
> V7 war die letzte Forschungsversion der Bell Labs, die breit lizenziert wurde. Aus ihr entstanden die kommerziellen Linien System III und System V von AT&T und, über Berkeley, die BSD-Systeme.

Zeitstrahl

Unix von 1969 bis heute

- **1969** — PDP-7-Unix
- **1971** — Unix V1
- **1973** — Unix V4: Kernel in C
- **1975** — Unix V6 an Hochschulen
- **1978** — 1BSD aus Berkeley
- **1979** — Unix V7 *(aktuell)*
- **1983** — 4.2BSD: TCP/IP, Sockets
- **1986** — 4.3BSD
- **1988** — POSIX-Standard
- **1991** — Linux 0.01
- **1993** — FreeBSD, NetBSD
- **2001** — Mac OS X (BSD-Wurzeln)
- **2025** — V4-Band in Utah gefunden
- **heute** — Linux, macOS, Android, BSD

1 Station in dieser Lernhilfe

Station 5: 4.3BSD (1986)

An der University of California in Berkeley entstand ab 1977 die Berkeley Software Distribution (BSD). 4.2BSD brachte 1983 den TCP/IP-Stack und die Socket-Schnittstelle, mit der Programme bis heute auf fast jedem Betriebssystem Netzwerkverbindungen aufbauen. 4.3BSD erschien 1986 und lief auf dem DEC VAX-11/780.

Starten:

```
docker run -it --name bsd43 -p 2323:2323 \
    -e SIMH_USE_CONTAINER=yes jguillaumes/simh-vaxbsd
```

Der Befehl ist für die Bash umbrochen, in PowerShell schreiben Sie ihn ohne `\` in eine Zeile. Beim allerersten Start prüft BSD das Dateisystem und der Container beendet sich. Das ist normal. Danach startet er mit:

```
docker start -ai bsd43
```

Mit Compose: `docker attach bsd43`. Den Neustart nach dem ersten Dateisystem-Check erledigt Docker dort automatisch.

Login: `root`, ohne Passwort. Der Prompt ist `vaxbsd#`.

Aufgaben

Befehl	Was Sie sehen
`hostname`, `uptime`	`vaxbsd`, dazu Laufzeit und Last wie bei Linux.
`ifconfig de0`	Die Netzwerkkarte de0 mit der IP-Adresse 172.17.0.103.
`netstat -rn`	Die Routing-Tabelle, Default-Gateway ist 172.17.0.1.
`ping 172.17.0.1 64 3`	Drei Antworten vom Docker-Host. Die Syntax ist noch alt: Ziel, Paketgröße, Anzahl.
`ping 8.8.8.8 64 3`	Ein Rechner von 1986 erreicht über die NAT von Docker das Internet.
`ls /usr/games`	Spiele von damals, zum Beispiel adventure, rogue und hangman.
`/usr/games/fortune`	Ein zufälliger Spruch, wie ihn viele Systeme beim Login zeigten.

Hinweis: Die VAX hat im Standardnetz von Docker die feste Adresse 172.17.0.103. Starten Sie deshalb pro Docker-Host nur einen BSD-Container.

Mini-Übung: Mehrbenutzerbetrieb wie 1986

Öffnen Sie ein zweites Terminal und melden Sie sich per Telnet an der VAX an:

```
telnet localhost 2323
```

Melden Sie sich als root an. Mit `who` sehen Sie, wer an welchem Terminal angemeldet ist.

Sauber herunterfahren:

```
sync; /etc/halt
```

Ohne halt prüft BSD beim nächsten Start wieder das Dateisystem. Den Container löschen Sie danach mit `docker rm -v bsd43`. Bei Compose startet Docker BSD nach halt automatisch neu, ganz anhalten können Sie es mit `docker compose stop bsd43`.

> **Info: Sockets von 1983 in Ihrem Alltag**
>
> Die Berkeley-Sockets sind die Grundlage fast aller Netzwerkprogramme. Ein Webserver, ping unter Linux oder das Python-Modul socket benutzen dieselben Aufrufe wie socket(), bind() und connect().

Zeitstrahl

Unix von 1969 bis heute

- **1969** PDP-7-Unix
- **1971** Unix V1
- **1973** Unix V4: Kernel in C
- **1975** Unix V6 an Hochschulen
- **1978** 1BSD aus Berkeley
- **1979** Unix V7
- **1983** 4.2BSD: TCP/IP, Sockets
- **1986** 4.3BSD
- **1988** POSIX-Standard
- **1991** Linux 0.01
- **1993** FreeBSD, NetBSD
- **2001** Mac OS X (BSD-Wurzeln)
- **2025** V4-Band in Utah gefunden
- **heute** Linux, macOS, Android, BSD

1 Station in dieser Lernhilfe

Rückblick: Was ist geblieben?

Vieles, was Sie unter Linux täglich benutzen, ist in den fünf Stationen entstanden:

Thema	Damals	Linux heute
Verzeichnis wechseln	`chdir`, ab V7 `cd`	`cd`
Rechte in ls -l	`sdrwr-` (V1), ab V4 `drwxr-xr-x`	`drwxr-xr-x`
Sprache des Kernels	Assembler bis V3, ab V4 C	C, inzwischen auch Rust
Übersetztes Programm	`a.out` seit PDP-7-Unix	`a.out` bei gcc ohne -o
Passwort-Hash	in /etc/passwd (V7)	in /etc/shadow
Shell-Schleifen	`for ... do ... done` (Bourne-Shell, V7)	gleiche Syntax in der Bash
Netzwerk	TCP/IP und Sockets ab 4.2BSD (1983)	TCP/IP und Sockets
Tippfehler	# und @	Rücktaste und Strg+U

Kontrollfragen

1. Warum war der Wechsel zu C bei Unix V4 so wichtig?
2. Mit welchem Befehl wechselte man bei V1 das Verzeichnis, und seit welcher Version gibt es cd?
3. Welche Sprache war der Vorgänger von C, und auf welchem System haben Sie sie ausprobiert?
4. Welche BSD-Version brachte TCP/IP, und wo begegnet Ihnen die Socket-Schnittstelle heute?
5. Wo stand der Passwort-Hash bei V7, und wo steht er unter Linux?

Quellen und Links

- Open SIMH: github.com/open-simh/simh
- PDP-7-Unix: github.com/DoctorWkt/pdp7-unix
- Unix V1: github.com/DoctorWkt/unix-jun72
- Unix V4 (Band aus Utah): www.tuhs.org/Archive/Distributions/Research/Utah_v4
- Unix V7 und 4.3BSD: hub.docker.com/u/jguillaumes
- The Unix Heritage Society: www.tuhs.org

Zeitstrahl

Unix von 1969 bis heute

- **1969** — PDP-7-Unix
- **1971** — Unix V1
- **1973** — Unix V4: Kernel in C
- **1975** — Unix V6 an Hochschulen
- **1978** — 1BSD aus Berkeley
- **1979** — Unix V7
- **1983** — 4.2BSD: TCP/IP, Sockets
- **1986** — 4.3BSD
- **1988** — POSIX-Standard
- **1991** — Linux 0.01
- **1993** — FreeBSD, NetBSD
- **2001** — Mac OS X (BSD-Wurzeln)
- **2025** — V4-Band in Utah gefunden
- **heute** — **Linux, macOS, Android, BSD**

1 Station in dieser Lernhilfe