



Universität Hamburg  
DER FORSCHUNG | DER LEHRE | DER BILDUNG

# POSIX

PROSEMINAR – Speicher- & Dateisysteme  
**POSIX – Portable Operating System Interface**  
Johnvir Singh Khattar – 18.12.2018  
Betreuer: Eugen Betke

## ***Gliederung***

- 1) **POSIX - Portable Operating System Interface**
  - i. Generelle Informationen – Was?, Wer?, Wo?
  - ii. Entwicklung – Versionsverlauf
  - iii. Begründung – Warum entwickelt?
- 2) **POSIX – Spezifikation**
  - i. Aufbau des Standards / der Spezifikation
  - ii. Beispielauszüge – Erläuterungen
  - iii. Ausdrücke
- 3) **POSIX – Verwendung & Relevanz**
- 4) **POSIX – Kritik**
- 5) **Fazit / Ausblick**

## Portable Operating System Interface

- Gruppe von Standards, Basiert auf Unix
  - Standardisierte Programmierschnittstelle (API)
  - Entwickelt durch: IEEE & Open Group für Unix
  - Schnittstelle: Anwendungssoftware & Betriebssystem
- Kompatibilität, Standard-Verhalten



[1]



[2]

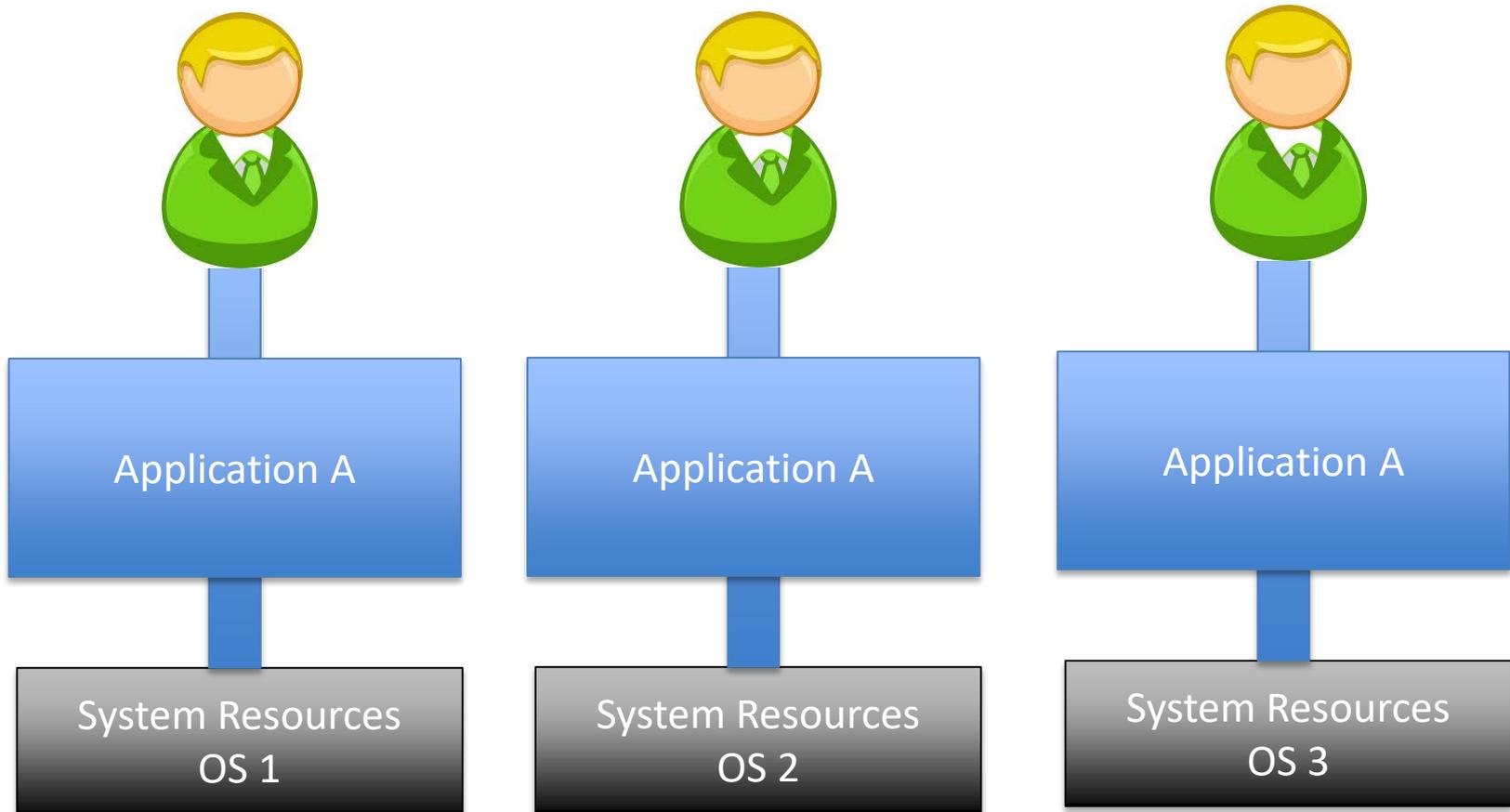
## Versionsverlauf

- POSIX.1 : IEEE Std 1003.1-1988 (Core Services: Process Creation & Control, C Library, File and Directory Operations, Signals[Floating Point Exceptions, Illegal instructions, memory violations, etc.] )
- POSIX.1b : IEEE Std 1003.1b-1993 (Real-time extensions: Shared memory, message passing, Real time signals, semaphores)
- POSIX.1c : IEEE Std 1003.1c-1995 (Threads extensions: Thread Creation, Thread Scheduling, etc.)
- POSIX.2: IEEE Std 1003.2-1992 (Shell Utilities: Command Interpreter, Utility Programs)

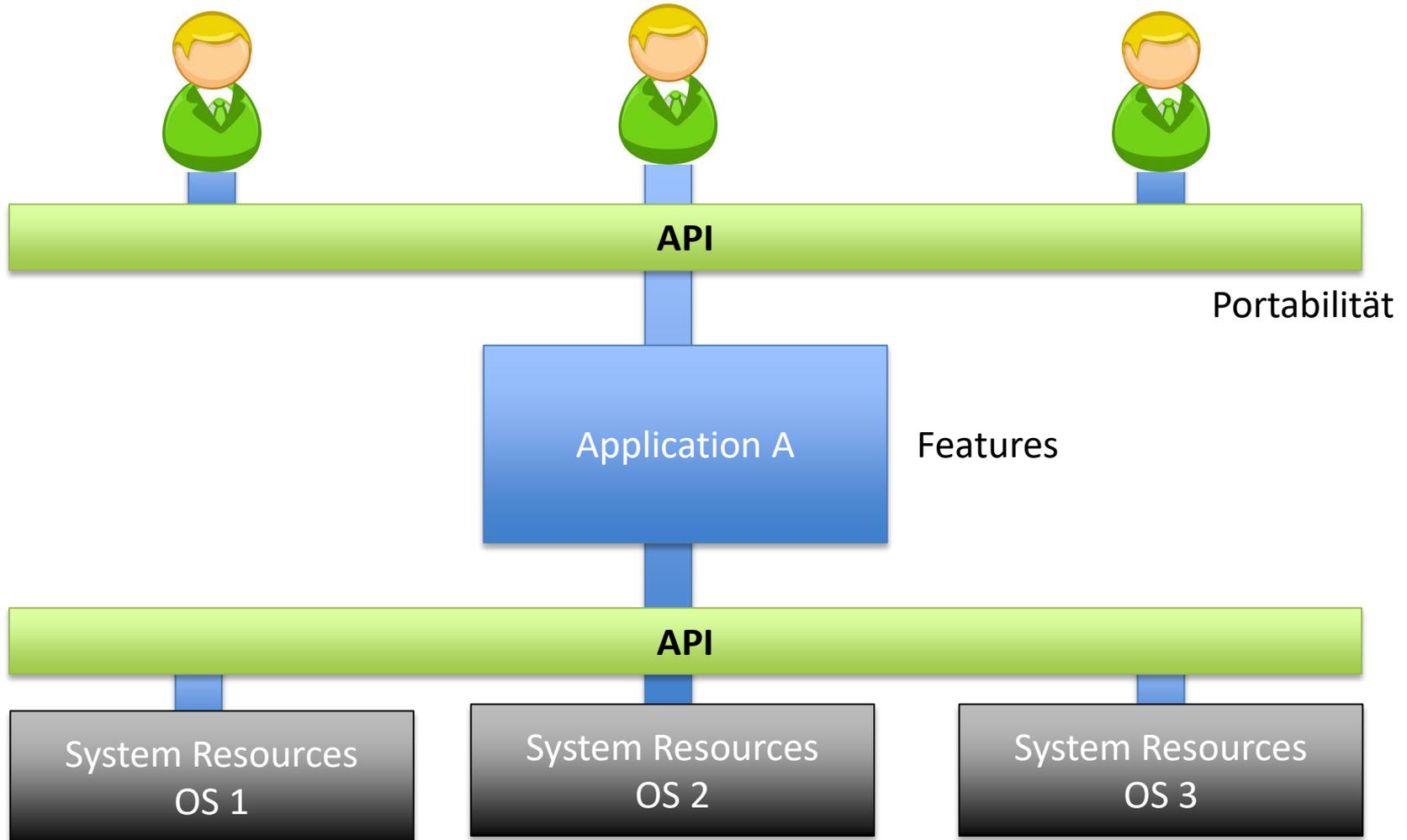
## Versionsverlauf

### AUSTIN GROUP Nach 1997

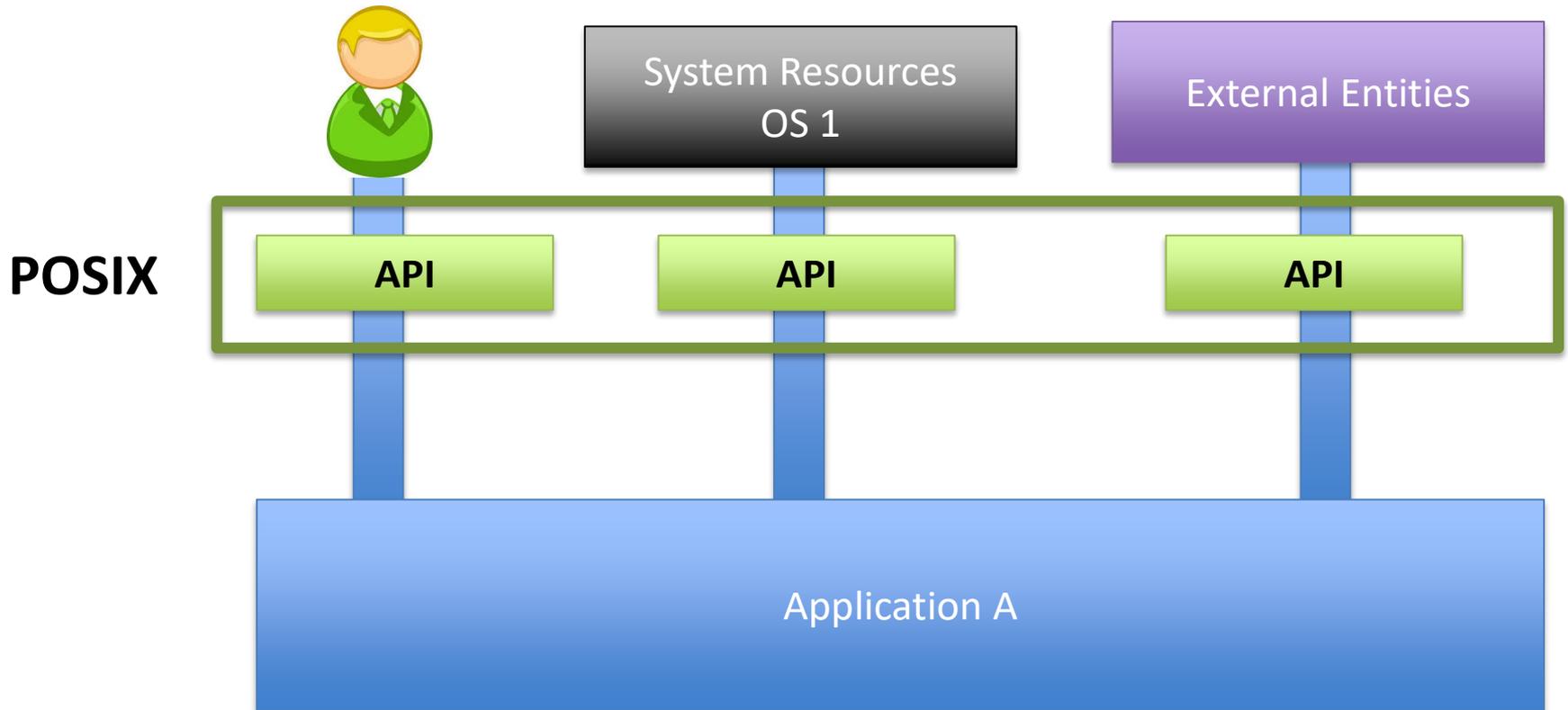
- POSIX.1-2001: Base Definitions, System Interfaces und Headers, Commands und Utilities
- POSIX.1-2004: Kleines Update, neue techn. Anforderungen
- POSIX.1-2008: Base Definitions, System Interfaces und Headers, Commands und Utilities, rationale volume
- POSIX.1-2017: Base specifications
- **Heutiger Standard: ISO/IEC/IEEE 9945, IEEE Std 1003.1-2008**



# POSIX Begründung 2/3



[3]



[3]

- IEEE Std 1003.1 (POSIX.1) – Prinzipien:

Prinzip	Erläuterung
Min. Interface, Not Implementation	Interface, ohne Implementation
Portability	Betriebssystemunabhängig
Application-oriented	Applikationsorientierung
Broadly Implementable	Einsetzbar über viele OS hinweg
Min. Changes to: Hist. Implementation & Exist. Application Code	Minimale Veränderungen der bestehenden Implementation & des bestehenden Anwendungscodes

- **Inhalt Spezifikation** von Standard IEEE Std 1003.1-2008:
  - Basis-Definitionen (Konventionen, Definitionen, Konzepte)
  - System-Schnittstelle (API, Standard C)
  - Erläuterungen

## POSIX Threads (Pthreads)

- Thread: Satz von Anweisungen (Prozedur)
- Multi-threaded Programme: Mehrere Prozeduren gleichzeitig
- Pthreads: [pthread.h](http://pthread.h) pthread\_create() / pthread\_join()
- Programm Performance
  - Weniger OS overhead
  - Weniger Ressourcen Verbrauch
  - Mehr Threads gleichzeitig möglich
- Unterstützt durch meisten OS

## POSIX Threads (Pthreads)

- Output:

0000011111

```

9  #include <stdio.h>
10 #include <pthread.h>
11
12 void *print_char (void *ch)
13 {
14     int i;
15     for (i=0; i<5; i++)
16         printf ("%c", *(char*)ch);
17     return NULL;
18 }
19
20 int main ()
21 {
22     char ch1='1', ch2='0';
23     pthread_t p1, p2;
24
25     pthread_create (&p1, NULL, print_char, &ch1);
26     pthread_create (&p2, NULL, print_char, &ch2);
27
28     pthread_join (p1, NULL);
29     pthread_join (p2, NULL);
30
31     printf ("\n");
32     return 0;
33 }

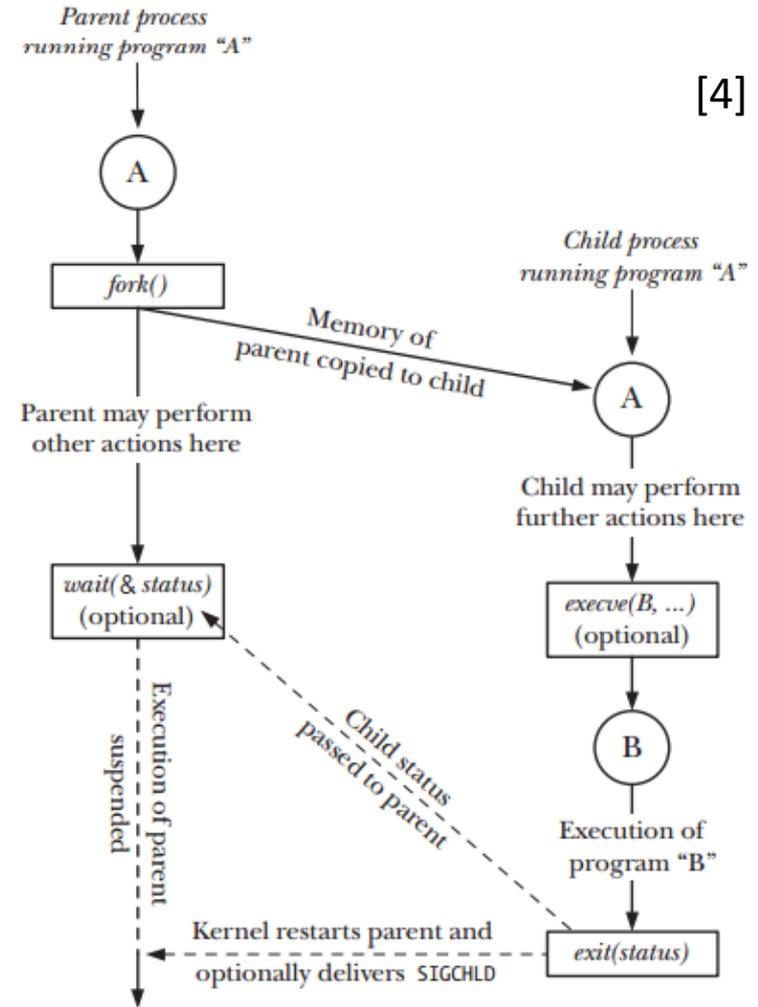
```

## POSIX Prozesse

- `fork()` mit `exec()` neuer Prozess
- `exit()` zurück zum Parent
- `kill()` Prozess beenden

### Interprocess-Komm.:

- Über system call `fork()`: -1, 0, pid
- [Signals](#)



## POSIX I/O bzw. Dateizugriff

- Beispiel mit `fopen()`/ `fwrite()` / `fread()` / `fclose()`
- Erstellung Datei `SDS.txt`
- Content: `SDS-Proseminar`

```

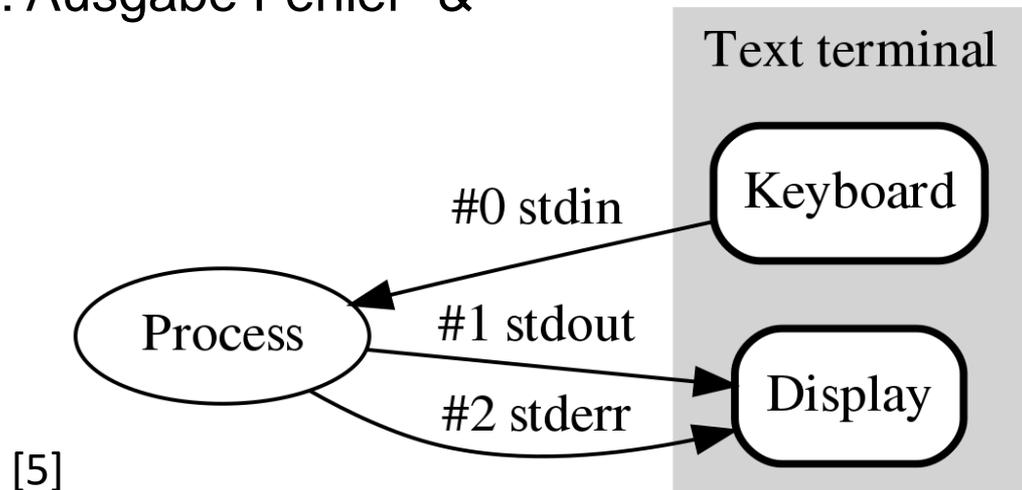
7  *****
8
9  #include<stdio.h>
10
11 int main () {
12     FILE *fp;
13     char str[] = "SDS-Proseminar";
14
15     fp = fopen( "SDS.txt" , "w" );
16     fwrite(str , 1 , sizeof(str) , fp );
17
18     fclose(fp);
19
20     return(0);
21 }
22

```

## Terminal I/O - Datenströme

### Standard-Dateien:

- stdin, Standardeingabe: Daten in Programm einlesen
- stdout, Standardausgabe: Daten aus Programm ausgeben
- stderr, Standardfehlerausgabe: Ausgabe Fehler- & Statusmeldungen



# POSIX Ausdrücke 1/2

POSIX	Description	ASCII	Unicode	Shorthand	Java
<code>[ :alnum: ]</code>	Alphanumeric characters	<code>[a-zA-Z0-9]</code>	<code>[\p{L}\p{N}]</code> <code>\p{Nd}</code>		<code>\p{A}num</code>
<code>[ :alpha: ]</code>	Alphabetic characters	<code>[a-zA-Z]</code>	<code>\p{L}\p{N}</code>		<code>\p{A}pha</code>
<code>[ :ascii: ]</code>	ASCII characters	<code>[\x00-\x7F]</code>	<code>\p{InBasicLatin}</code>		<code>\p{ASCII}</code>
<code>[ :blank: ]</code>	Space and tab	<code>[ \t]</code>	<code>[\p{Zs}\t]</code>	<code>\h</code>	<code>\p{B}lank</code>
<code>[ :cntrl: ]</code>	Control characters	<code>[\x00-\x1F\x7F]</code>	<code>\p{Cc}</code>		<code>\p{C}ntrl</code>
<code>[ :digit: ]</code>	Digits	<code>[0-9]</code>	<code>\p{Nd}</code>	<code>\d</code>	<code>\p{D}igit</code>
<code>[ :punct: ]</code>	Punctuation (and symbols).	<code>[! " \# \$% &amp; ' ( ) * + ,</code> <code>\- . / : ; &lt; = &gt; ? @ [ \</code> <code>\\ ] ^ _ ` {   } ~ ]</code>	<code>\p{P}</code>		<code>\p{P}unct</code>
<code>[ :space: ]</code>	All whitespace characters, including line breaks	<code>[ \t\r\n\v\f]</code>	<code>[\p{Z}\t\r\n\v\f]</code>	<code>\s</code>	<code>\p{S}pace</code>
<code>[ :upper: ]</code>	Uppercase letters	<code>[A-Z]</code>	<code>\p{Lu}</code>	<code>\u</code>	<code>\p{U}pper</code>

## C-Posix Library

- Spezifikation einer Standard C Bibliothek
- POSIX: Mehr Funktionen, als Standard C
- [Publikation Open Group POSIX headers](#)

## OS - POSIX konform

A/UX

macOS mit Darwin

Solaris

AIX

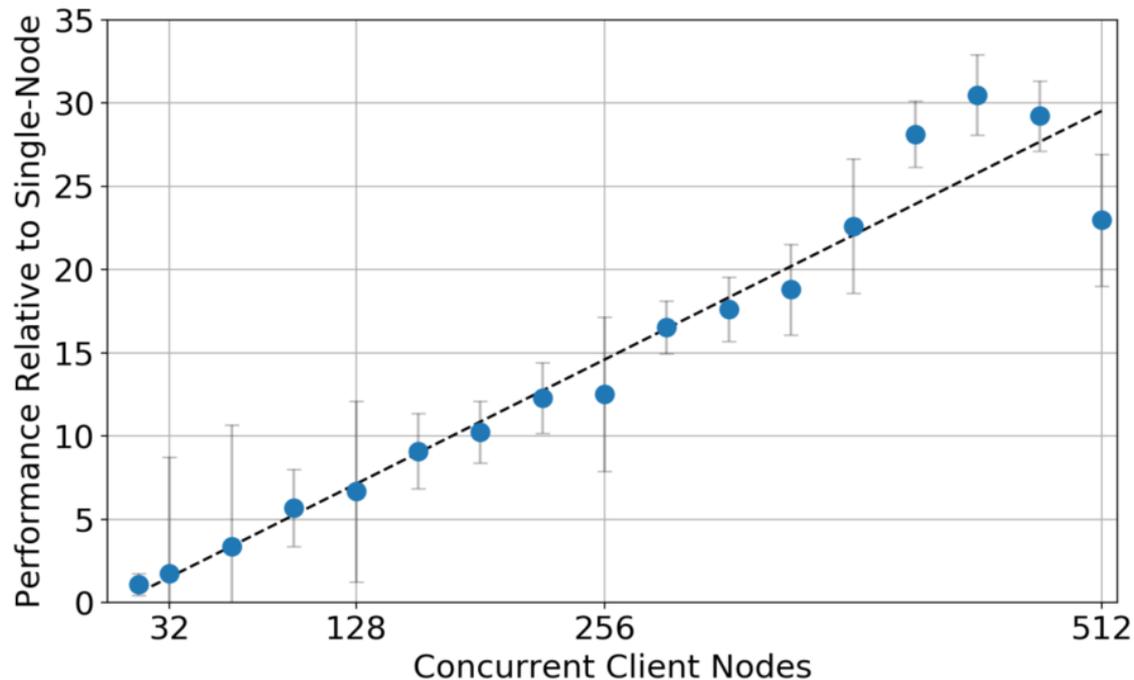
HP-UX

Windows NT (Pthreads-w32)

LINUX

- Zur Einführung große Relevanz & Reichweite
- Größter Standard für OS auf dem Markt
- Kritik steigt → veraltet?
  - 3 große UNIX-OS:  
Android, OS X, Ubuntu
- [Google Trends](#)

Performance of Concurrent File Opens



[7]

x-Achse: Anzahl der parallelen Klienten-Anfragen

Y-Achse: Performance Relativ zu Abfrage durch einen Klienten

*„After a **write()** to a regular file has successfully returned:*

- Any successful `read()` from each byte position in the file that was modified by that write shall return the data specified by the `write()` for that position until such byte positions are again modified.*
- Any subsequent successful `write()` to the same byte position in the file shall overwrite that file data.“ [Q.1]*

## POSIX Metadata sieht vor:

- User and group which owns the file
- Permission that user and group has to read and modify the file
- The time the file was created and last modified
- Etc.

- Ziel von POSIX: Portabilität von AS sichern.
- Umsetzung: Erstellung einiger Standards (API) für die OS-Hersteller → Garantiertes Verhalten
- Elemente von POSIX bis heute relevant: pthreads
- Heute: Kritik → POSIX-Standard sollte neu aufgesetzt werden + „outdated“ für neue Technologien



---

# Danke für Eure Aufmerksamkeit!

## Fragen?



# POSIX Abbildungsverzeichnis

- [1] Logo IEEE ; <https://www.ieee.org/>
- [2] Logo Open Group ; <https://www.opengroup.org/>
- [3] In Anlehnung an: „Understanding the POSIX Open System Reference Model (1990), Tom Elliot/BMI, <https://www.youtube.com/watch?v=31bS6cUHj-U>
- [4] Übernommen aus: „Linux Processes and Signals“, [https://www.bogotobogo.com/Linux/linux\\_process\\_and\\_signals.php](https://www.bogotobogo.com/Linux/linux_process_and_signals.php)
- [5] Übernommen aus: „Standard-Datenströme“, <https://de.wikipedia.org/wiki/Standard-Datenstr%C3%B6me>
- [6] Gekürzt übernommen aus: „POSIX Bracket Expressions“ , <https://www.regular-expressions.info/posixbrackets.html>
- [7] Übernommen aus: Glenn Lockwood, „WHAT´S SO BAD ABOUT POSIX I/O?“ , <https://www.nextplatform.com/2017/09/11/whats-bad-posix-io/>



[Q.1] <http://pubs.opengroup.org/onlinepubs/9699919799/functions/write.html>  
<http://pages.cs.wisc.edu/~remzi/OSTEP/>  
<https://kb.iu.edu/d/agjv>  
[https://www.unidata.ucar.edu/mailling\\_lists/archives/lm-users/2002/msg00512.html](https://www.unidata.ucar.edu/mailling_lists/archives/lm-users/2002/msg00512.html)  
<http://www.open-std.org/jtc1/sc22/WG15/>  
<https://www.regular-expressions.info/posixbrackets.html>  
<https://de.slideshare.net/djhseen1993/posix-46606116>  
[http://www.opengroup.org/austin/papers/posix\\_faq.html](http://www.opengroup.org/austin/papers/posix_faq.html)  
<http://sourceware.org/pthreads-win32/>  
<https://www.nextplatform.com/2017/09/11/whats-bad-posix-io/>  
[https://de.wikipedia.org/wiki/Open\\_Group](https://de.wikipedia.org/wiki/Open_Group)  
<https://en.wikipedia.org/wiki/POSIX>  
[https://de.wikipedia.org/wiki/Portable\\_Operating\\_System\\_Interface](https://de.wikipedia.org/wiki/Portable_Operating_System_Interface)  
[https://de.wikipedia.org/wiki/Native\\_POSIX\\_Thread\\_Library](https://de.wikipedia.org/wiki/Native_POSIX_Thread_Library)  
<http://pubs.opengroup.org/onlinepubs/9699919799.2016edition/>  
[https://en.wikipedia.org/wiki/Du\\_\(Unix\)](https://en.wikipedia.org/wiki/Du_(Unix))  
[https://en.wikipedia.org/wiki/Df\\_\(Unix\)](https://en.wikipedia.org/wiki/Df_(Unix))  
<http://www.opengroup.org/austin/papers/backgrounder.html>  
<https://de.wikipedia.org/wiki/Standard-Datenstr%C3%B6me>



# POSIX Literaturverzeichnis

<http://www.cs.columbia.edu/~vatlidak/resources/POSIXmagazine.pdf>

<http://pages.cs.wisc.edu/~remzi/OSTEP/cpu-mechanisms.pdf>

<http://pages.cs.wisc.edu/~remzi/OSTEP/threads-api.pdf>

<http://pages.cs.wisc.edu/~remzi/OSTEP/threads-locks.pdf>

<http://pages.cs.wisc.edu/~remzi/OSTEP/threads-cv.pdf>

<http://pages.cs.wisc.edu/~remzi/OSTEP/threads-sema.pdf>

<http://pages.cs.wisc.edu/~remzi/OSTEP/file-devices.pdf>

<http://pages.cs.wisc.edu/~remzi/OSTEP/dist-nfs.pdf>

<http://pages.cs.wisc.edu/~remzi/OSTEP/dialogue-monitors.pdf>

<http://pages.cs.wisc.edu/~remzi/OSTEP/threads-monitors.pdf>

<http://pages.cs.wisc.edu/~remzi/OSTEP/lab-projects-systems.pdf>

<https://www.youtube.com/watch?v=U0GbJtnfqSM>

<https://www.youtube.com/watch?v=hy4OeVCLGZ4>

<https://www.youtube.com/watch?v=QihYQ743k4A>

<https://www.youtube.com/watch?v=31bS6cUHj-U>

<https://www.youtube.com/watch?v=8ZTGmsi9U9U>

<http://www.ijon.de/comp/tutorials/threads/create.html>

[https://www.bogotobogo.com/Linux/linux\\_process\\_and\\_signals.php](https://www.bogotobogo.com/Linux/linux_process_and_signals.php)

[https://www.tutorialspoint.com/c\\_standard\\_library/c\\_function\\_fwrite.htm](https://www.tutorialspoint.com/c_standard_library/c_function_fwrite.htm)