This Page Intentionally Left Blank Space

LIVE FREE OR DIE

UNIX*

TRADEMARK OF BELL LABS*

Unix-like systems

Year		
1969		
1971 to 1973		
1974 to 1975		
1978		
1979		
1980		
1981		
1982		
1983		
1984		
1985		
1986		
1987		
1988		
1989		
1990		
1991		
1992		
1993		
1994		
1995		
1996		
1997		
1998		
1999		
2000		
2001 to 2002		
2003		
2004		
2005 to 2007		
2008 to 2009		
2010		
2011 to 2018		
2019 to 2023		

Legend:
- Open source
- Mixed/shared source
- Closed source
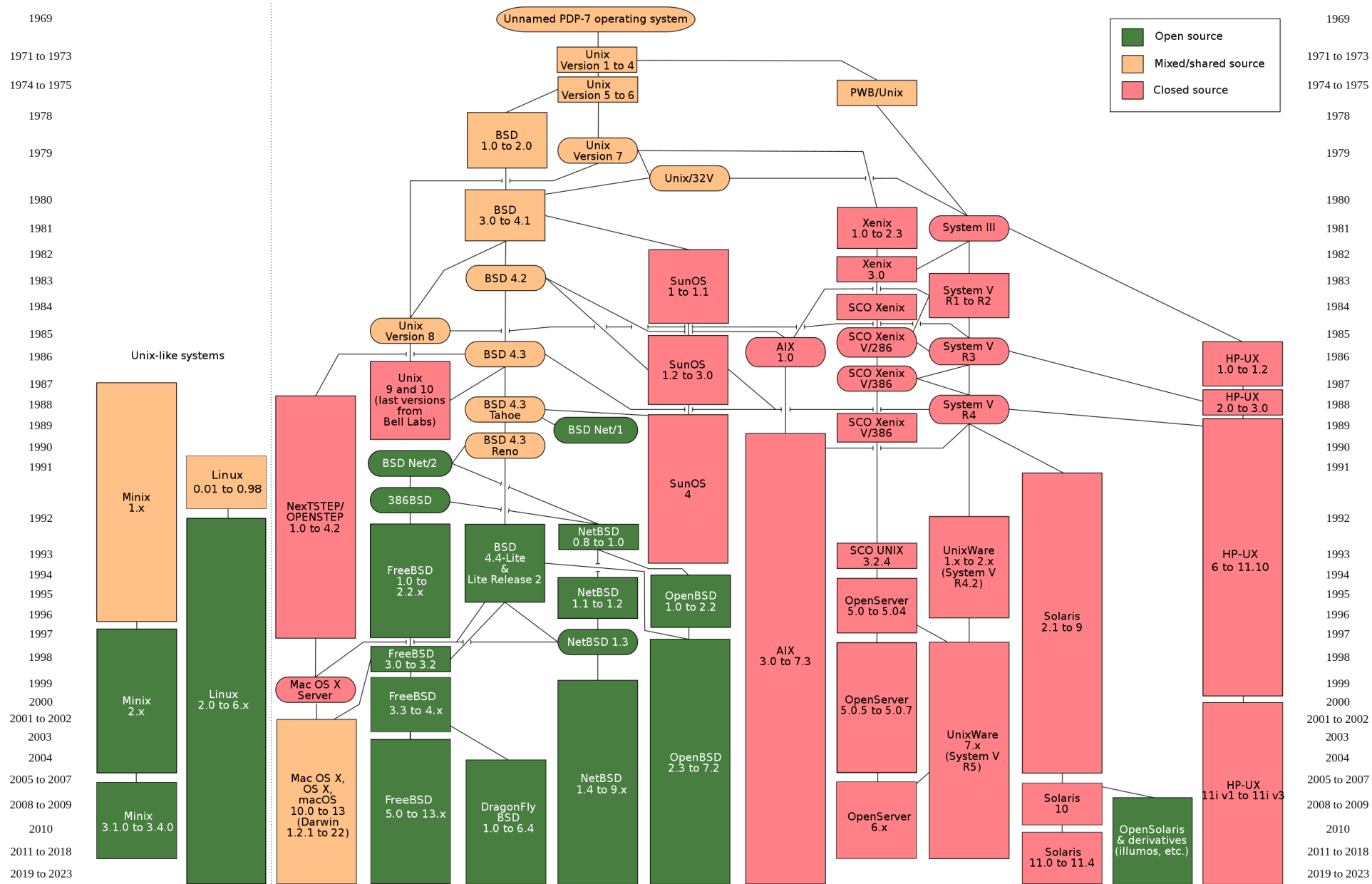
The Single UNIX® Specification History & Timeline

Year		Description
1969	The Beginning	The history of UNIX starts back in 1969, when Ken Thompson, Dennis Ritchie and others started working on the "little-used PDP-7 in a corner" at Bell Labs and what was to become UNIX.
1971	First Edition	It had an assembler for a PDP-11/20, file system, fork(), roff and ed. It was used for text processing of patent documents.
1972	First UNIX Installations	The first installations had 3 users, no memory protection, and a 500 KB disk.
1973	Fourth Edition	It was rewritten in C. This made it portable and changed the history of OS's.
1975	Sixth Edition	UNIX leaves home. Also widely known as Version 6, this is the first to be widely available outside of Bell Labs. The first BSD version (1.x) was derived from V6.
1979	Seventh Edition	It was an "improvement over all preceding and following Unices" [Bourne]. It had C, UUCP and the Bourne shell. It was ported to the VAX and the kernel was more than 40 Kilobytes (K).
1980	Xenix	Microsoft introduces Xenix. 32V and 4BSD introduced.
1982	System III	AT&T's UNIX System Group (USG) release System III, the first public release outside Bell Laboratories. SunOS 1.0 ships. HP-UX introduced. Ultrix-11 introduced.
1983	System V	Computer Research Group, UNIX System Group (USG) and a third group merge to become UNIX System Development Lab. AT&T announces UNIX System V, the first supported release. Installed base 45,000.
1984	4.2BSD	University of California at Berkeley releases 4.2BSD, includes TCP/IP, new signals and much more. X/Open formed.
1984	SVR2	System V Release 2 introduced. At this time there are 100,000 UNIX installations around the world.
1986	4.3BSD	4.3BSD released, including internet name server. SVID introduced. NFS shipped. AIX announced. Installed base 250,000.
1987	SVR3	System V Release 3 including STREAMS, TLI, RFS. At this time there are 750,000 UNIX installations around the world. IRIX introduced.
1988		POSIX.1 published. Open Software Foundation (OSF) and UNIX International (UI) formed. Ultrix 4.2 ships.
1989		AT&T UNIX Software Operation formed in preparation for spinoff of UNIX development group. Motif 1.0 ships.
1989	SVR4	UNIX System V Release 4 ships, unifying System V, BSD and Xenix. Installed base 1.2 million.
1990	XPG3	X/Open launches XPG3 Brand. OSF/1 debuts. Plan 9 from Bell Labs ships.

"The Single UNIX Specification brings all the benefits of a single standard operating system, namely application and information portability, scalability, flexibility and freedom of choice for customers"

Allen Brown, President and CEO, The Open Group

The Story of the License Plate...

In 1983 Digital Equipment Corporation (DEC) was ramping up their engineering group to create and ship their first UNIX system product. One of the stalwarts of the group was Armando P. Stettner. Armando was a UNIX system devotee. He lived and he breathed the UNIX system. When he got his new car, it was natural that he got vanity license plates that said "UNIX" on them. And it only made it better that the state motto of New Hampshire was "Live Free or Die".

Armando often got requests from people along the lines of "When will we be able to get our UNIX system license directly from DEC?" And Armando kept saying "Real Soon Now" (RSN).

Armando was going to a conference and he was dreading having to say "RSN" many more times, so he had a bright idea. Amando went prepared to give out "UNIX licenses". On stage, when the question came up, "When will we be able to get our UNIX system license directly from DEC?", Armando yelled "Right Now!" and produced facsimilies of his license plate, holding it up for all to see. It was an almost perfect likeness of his license plate, with the trademark "UNIX" in the middle of it, but instead of having "Live Free or Die" across the bottom of the plate (as in the real case), it had it across the top. Across the bottom was the trademark acknowledgement.

DEC made UNIX license plates up in small numbers and handed them out at events. They usually ran out. The demand for the license plates never did abate. People saw them on an office wall, or heard about them somewhere, and wanted one of their own.

Armando left the state for the sunny climes of California, and had taken his car and license plate with him. Or so many people thought.

In 1989 Jon 'maddog' Hall was purchasing a new car, a Jeep Wrangler. And of course the license plate had to be relevant. So Jon, a long time DEC employee and UNIX system guru, submitted his application with many variations and the clerk said "I think we can give you your first choice...." and gave him the temporary paper plates (to be used on the car until the metal plates were manufactured) with "UNIX" on them. And so it has been ever since. Jon's Jeep has been the holder of the UNIX license plate.

The Open Group thanks Jon "maddog" Hall for sharing the story of the UNIX license plate.

The UNIX Brand

The UNIX Brand is used to identify products that have been certified as conforming to the Single UNIX Specification, initially UNIX 93, followed subsequently by UNIX 95, UNIX 98 and now UNIX 03.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Year		Description
1991		UNIX System Laboratories (USL) becomes a company - majority-owned by AT&T. Linus Torvalds commences Linux development. Solaris 1.0 debuts.
1992	SVR4.2	USL releases UNIX System V Release 4.2 (Destiny). October - XPG4 Brand launched by X/Open. December 22nd - Novell announces intent to acquire USL. Solaris 2.0 and HP-UX 9.0 ship.
1993	4.4BSD	4.4BSD the final release from Berkeley. June 16 - Novell acquires USL
Late 1993	SVR4.2MP	Novell decides to get out of the UNIX business. Rather than sell the business as a single entity, Novell transfers the rights to the UNIX trademark and the specification to X/Open Company. COSE Initiative delivers "Spec 1170" to X/Open for fasttrack. In December Novell ships SVR4.2MP , the final USL OEM release of System V
1994	Single UNIX Specification	BSD 4.4-Lite eliminated all code claimed to infringe on USL/Novell. As the owner of the UNIX trademark, X/Open introduces the Single UNIX Specification (formerly Spec 1170) which separates the UNIX trademark from any actual code stream itself, thus allowing multiple implementations.
1995	UNIX 95	X/Open introduces the UNIX 95 branding program for implementations of the Single UNIX Specification. Novell sells UnixWare business to SCO. Digital UNIX introduced. UnixWare 2.0 ships. OpenServer 5.0 debuts.
1996		The Open Group forms as a merger of the Open Software Foundation (OSF) and X/Open. UnixWare 2.1, HP-UX 10.20 and IRIX 6.2 ship.
1997	Single UNIX Specification, Version 2	The Open Group introduces Version 2 of the Single UNIX Specification, including support for realtime, threads and 64-bit and larger processors. The specification is made freely available on the web. IRIX 6.4, AIX 4.3 and HP-UX 11 ship.
1998	UNIX 98	The Open Group introduces the UNIX 98 family of brands, including Base, Workstation and Server. First UNIX 98 registered products shipped by Sun, IBM and NCR. The Open Source movement starts to take off with announcements from Netscape and IBM. UnixWare 7 and IRIX 6.5 ship.
1999	UNIX at 30	The UNIX system reaches thirty. Solaris 7 ships. Linux 2.2 kernel released. The Open Group and the IEEE commence joint development of a revision to POSIX and the Single UNIX Specification. First LinuxWorld conferences. Dot com fever on the stock markets. Tru64 UNIX ships.
2001	Single UNIX Specification, Version 3	Version 3 of the Single UNIX Specification unites IEEE POSIX, The Open Group and the industry efforts. Linux 2.4 kernel released. The value of procurements of open systems referencing the UNIX brand exceeds $55 billion. AIX 5L ships.
2003	ISO/IEC 9945	The core volumes of Version 3 of the Single UNIX Specification are approved as an international standard. "Westwood" test suites shipped for UNIX 03 brand. Solaris 9.0 E ships. Linux 2.6 kernel released.

THE OPEN GROUP

Making standards work™

LIVE FREE OR DIE

LINUX

LINUX is a Registered Trademark of Linus Torvalds

96B-0357

Most Popular *Linux Distributions*

		0	1000	2000
#1	Ubuntu			
#2	Fedora		1565	
#3	Mint		1415	
#4	openSUSE		1355	
#5	Debian		1037	
#6	Mandriva		1014	
#7	PCLinuxOS	794		
#8	Puppy	776		
#9	Sabayon	771		
#10	Arch	707		
#11	CentOS	618		
#12	MEPIS	579		
#13	Slackware	576		
#14	FreeBSD	542		
#15	Tiny Core	424		

Values: Hits Per Day in Distrowatch

DATA BROZ

Source: distrowatch

https://www.youtube.com/watch?v=YMCylaT4iV4

- American Telephone Company, AT&T:

  ✳Telefooncentrales produceerden tabellen die met de hand verwerkt moesten worden

- Bell Labs (AT&T's research laboratorium):

  ✳Produceerden nieuwe vindingen, 2 octrooi aanvragen per dag

  - Had de nieuwste DEC PDP 7 *mini computer* ($72.000, 500 KG)

    - Inspiratie voor Ken Thompson, Dennis Ritchie, Douglas McIlroy en Joe Ossanna om een nieuw operating systeem te ontwerpen: Uniplexed Information and Computing Service: UNICS, in 1970 omgedoopt naar UNIX.

    ✳Focus qua gebruik: tekst bewerken


```
"...the number of UNIX installations has grown to 10, with more expected..."
```
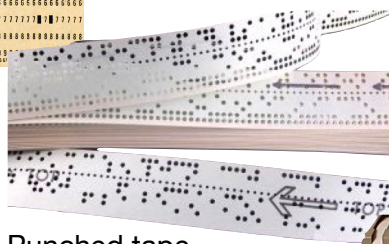*- Dennis Ritchie and Ken Thompson, June 1972*

More history: https://www.youtube.com/watch?v=EY6q5dv_B-o

Input & Output
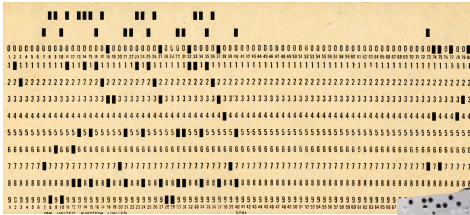
Ponskaart

Punched tape

NUL

CR   13/015/0xd

LF   10/012/0xa

DEL 127/0177/0x7f

SP   32/040/0x20 (&#x20)

"Delete" along with
some other ASCII
control characters
and space as they
appear on punched
tape

What is
TELETYPE?

ca. 1957
Teletype (TTY)

Terminal (command line)
ADM3a
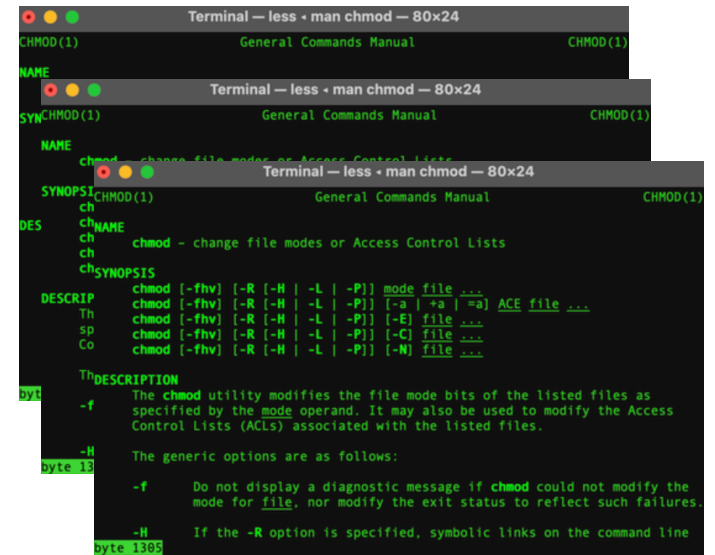
X11-Windows: Virtuel terminals

Directory a.k.a. map, folder	Files	Text Tools	Text Manipulation	Network Tools		Processes		Devices	Users	Shell
mkdir rmdir	ls cp rm	echo cat sort	ed vi(m) pico nano	ping ifconfig		ps kill		/dev /dev/tty… /dev/disk…	su	sh bash
pwd cd . cd .. mount	df du	wc od diff	sed tr dd	netstat	lsof			/dev/null /dev/zero	chmod	zsh ksh
/bin /sbin /usr/bin /etc/ /home	find tar	grep more less	formatting: nroff/man troff/man latex	tcpdump		init		dmesg	motd wall	csh

Vlaggetjes en Argumenten

		`-t`	sorteer op verandertijdtijd	
`ls`	sorteert	`-u`	sorteer op laatste leestijd	`ls -lt file1 file2`
list files	alfabetisch	`-a`	show (verborgen) punt-files	`ls -l -t file1 file2`

	telt bytes,	`-w`	tel woorden	
`wc`	karakters,	`-wl`	tel woorden en regels	`wc -l file1 file2`
count words	woorden en	`-c`	tel bytes	
	regels	`-m`	tel karakters	

`ps`				`ps 123`
list processes	list (jouw)	`ps -ef`	list alle processen (SYSV style)	list alleen proces met pid 123
	processen	`ps aux`	list alle processen (BSD style)	(process id)
`kill`				`kill -2 123` ^C op toetsenbord
stops processes				`kill -9 123` hardste kill

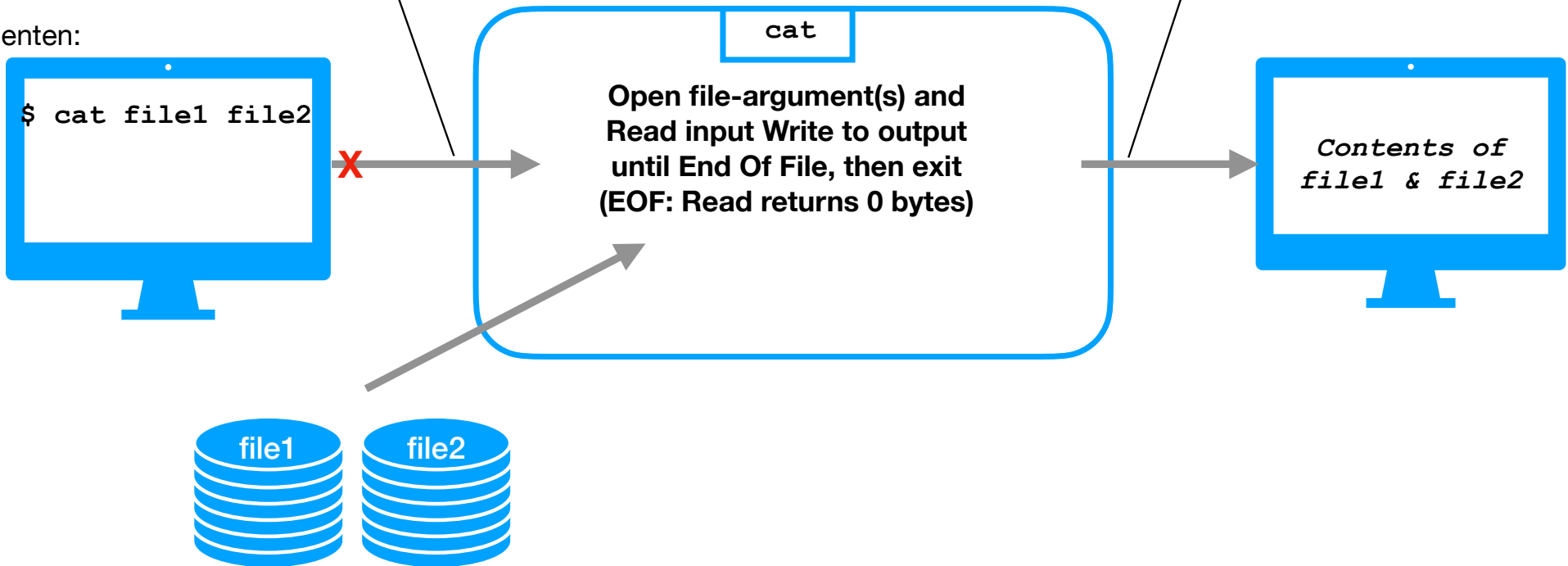soms is het een beetje vreemd

Input *of* Argumenten met de text utilities

- `cat` ←——— zonder argumenten

  - leest de invoer, standaard is dat het toetsenbord: *standard input*

- `cat file1 [file2 …]`

  - leest de bestanden die worden opgegeven als argumenten

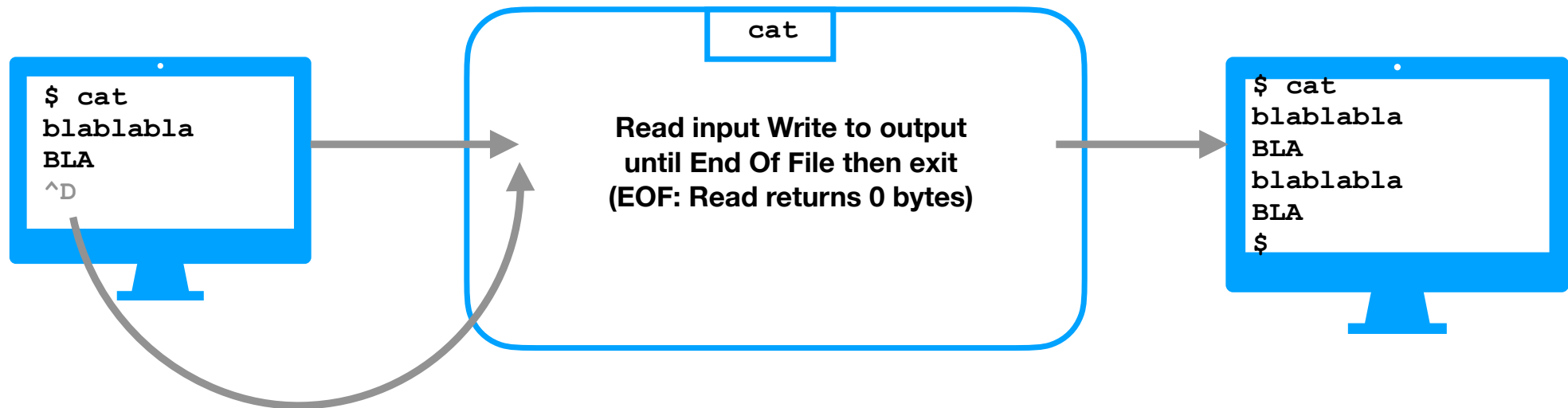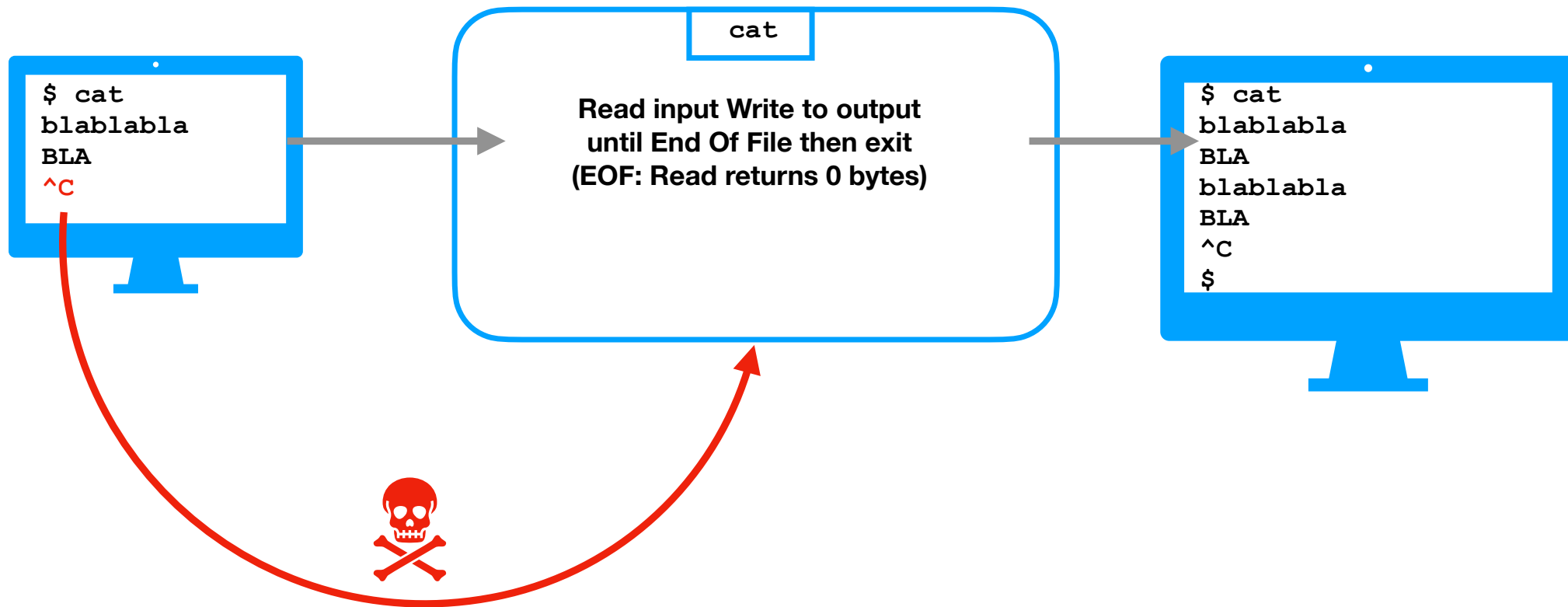 - standard input is er wel, maar wordt niet gebruikt voor invoer

Zonder argumenten:

```
$ cat
blablabla
BLA
```

cat

**Read input Write to output
until End Of File, then exit
(EOF: Read returns 0 bytes)**

```
$ cat
blablabla
BLA
blablabla
BLA
```

standard input (0)

standard output (1)

Met argumenten:

```
$ cat file1 file2
```

X

cat

**Open file-argument(s) and
Read input Write to output
until End Of File, then exit
(EOF: Read returns 0 bytes)**

*Contents of
file1 & file2*

file1 file2
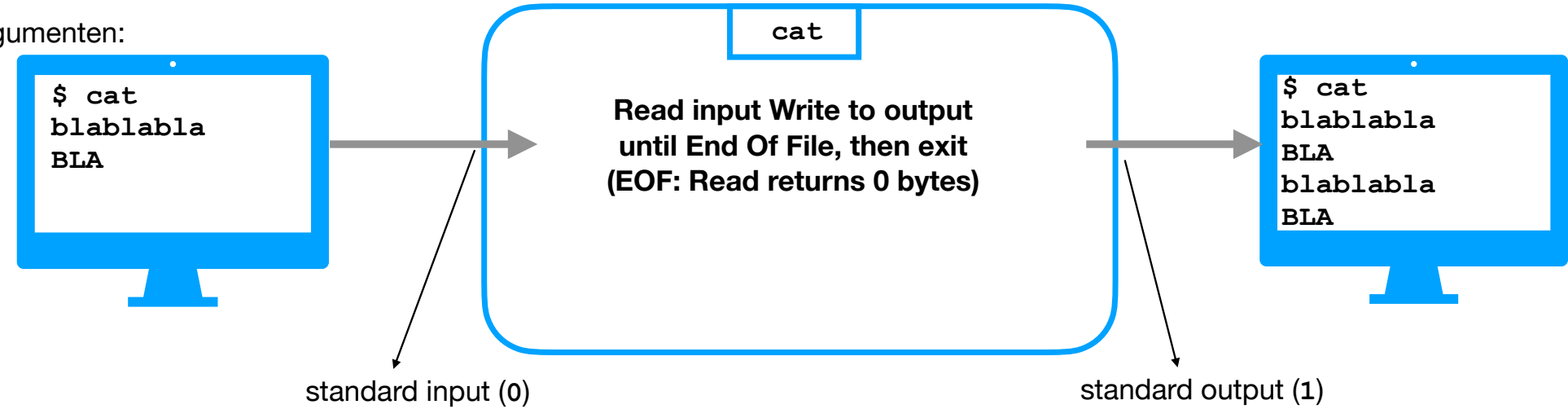
Input via de terminal stoppen: ^D

```
$ cat
blablabla
BLA
^D
```

cat

Read input Write to output until End Of File then exit (EOF: Read returns 0 bytes)

```
$ cat
blablabla
BLA
blablabla
BLA
$
```

Kernel terminal device driver translates keyboard ^D to: kernel sends 0 bytes (EOF)

Lopend programma afbreken via de terminal: `^C`

```
$ cat
blablabla
BLA
^C
```

```
cat
```
Read input Write to output until End Of File then exit (EOF: Read returns 0 bytes)

```
$ cat
blablabla
BLA
blablabla
BLA
^C
$
```

Kernel terminal device driver translates keyboard `^C` to: `kill -2` (terminal interrupt signal)

standard output and standard error output
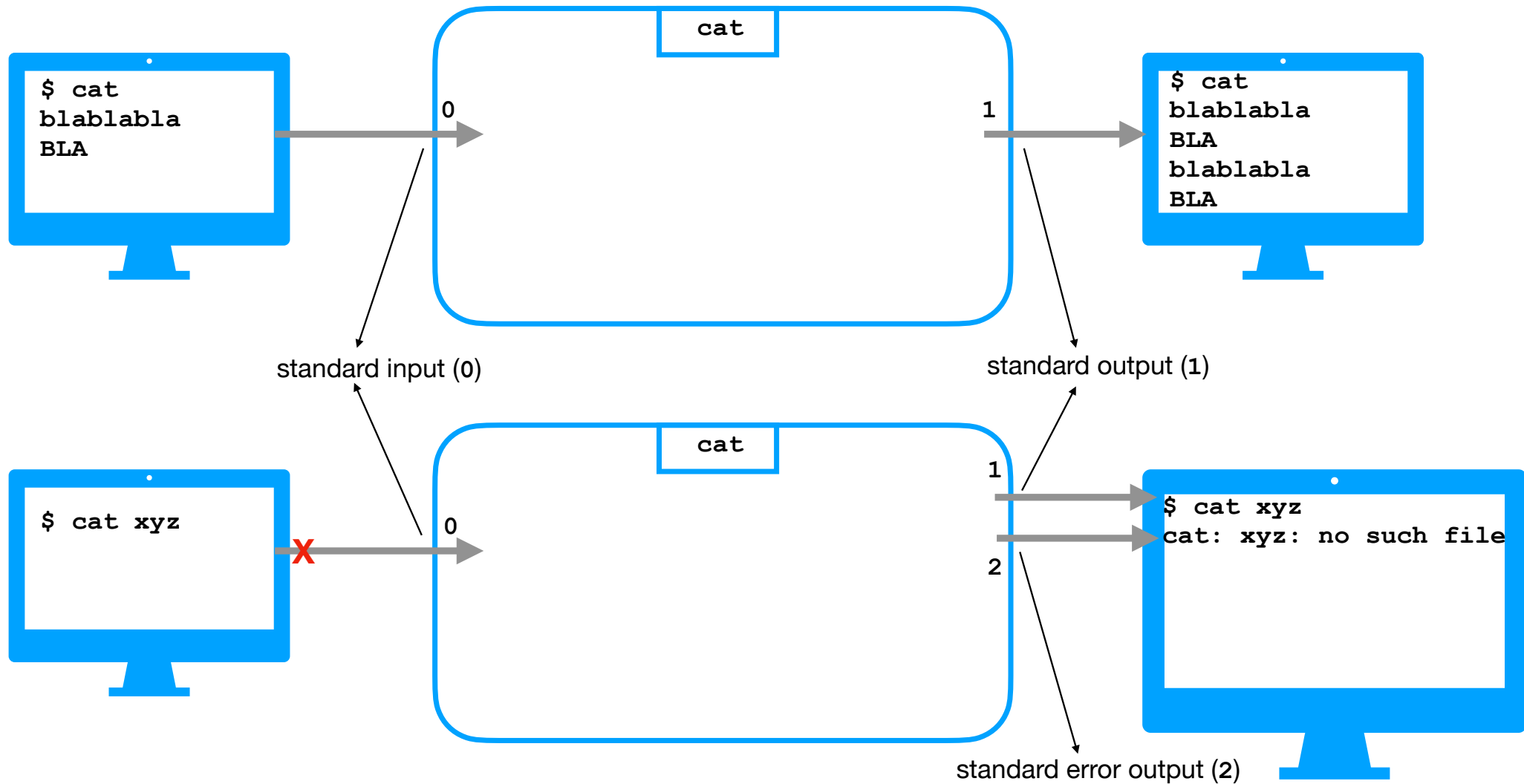
I/O Redirection

`$ cat 1>file`
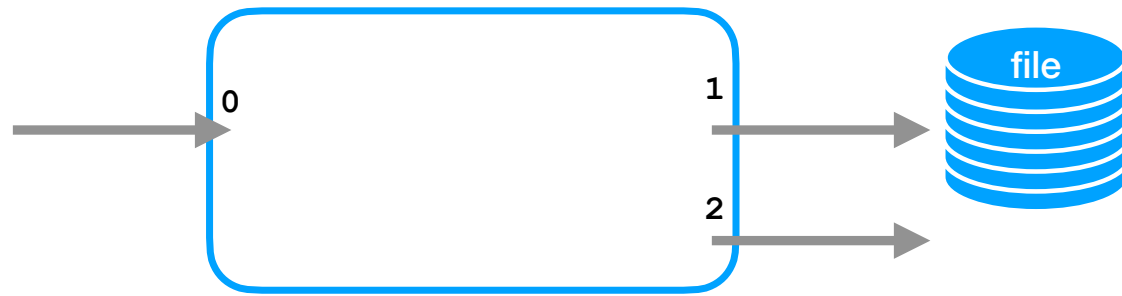
`$ cat  >file`

`$ cat 2>vang_fouten_op_file`

`$ cat 0<lees_uit_bestand`
`$ cat  <lees_uit_bestand`

`$ cat <input-file >output-file 2>error-output-file`
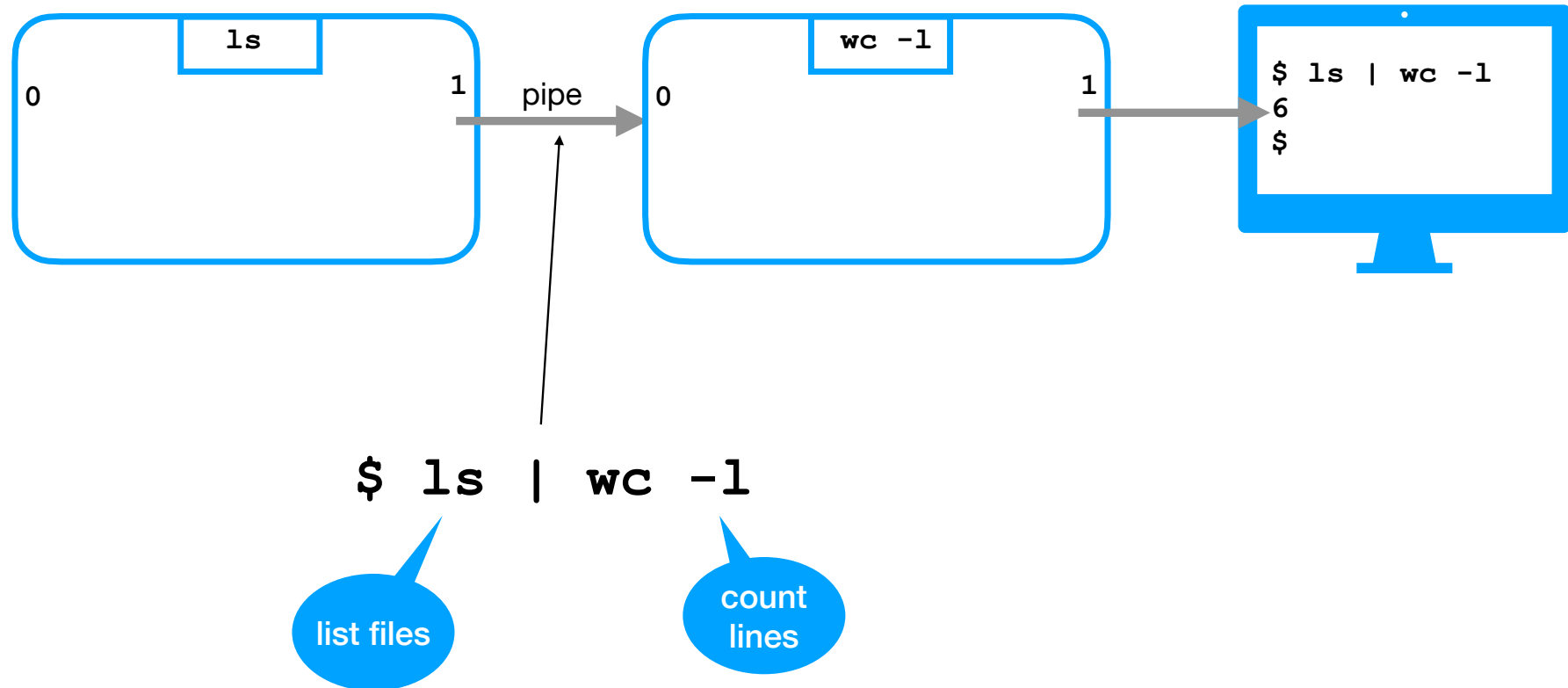
I/O Redirection

Combineer 2 met 1:

```
$ cat 2>&1          close(2); dup(1);          /* merge 2 with 1 */
```


Append:

```
$ cat >>output-file
```

I/O Redirection: pipes


```
        ls                         wc -1

0                1   pipe   0              1

                                    $ ls | wc -l
                                    6
                                    $
```

$ ls | wc -l

list files

count lines

I/O Redirection

Tel hoeveel files met aan letter **a** beginnen:

```
ls a* | wc -l
6
```

Stel dat er geen filenamen zijn die met een **a** beginnen, dan geeft **ls** een foutmelding:

```
ls a* | wc -l
ls: a*: No such file or directory
0
```

Onderdruk de foutmelding:
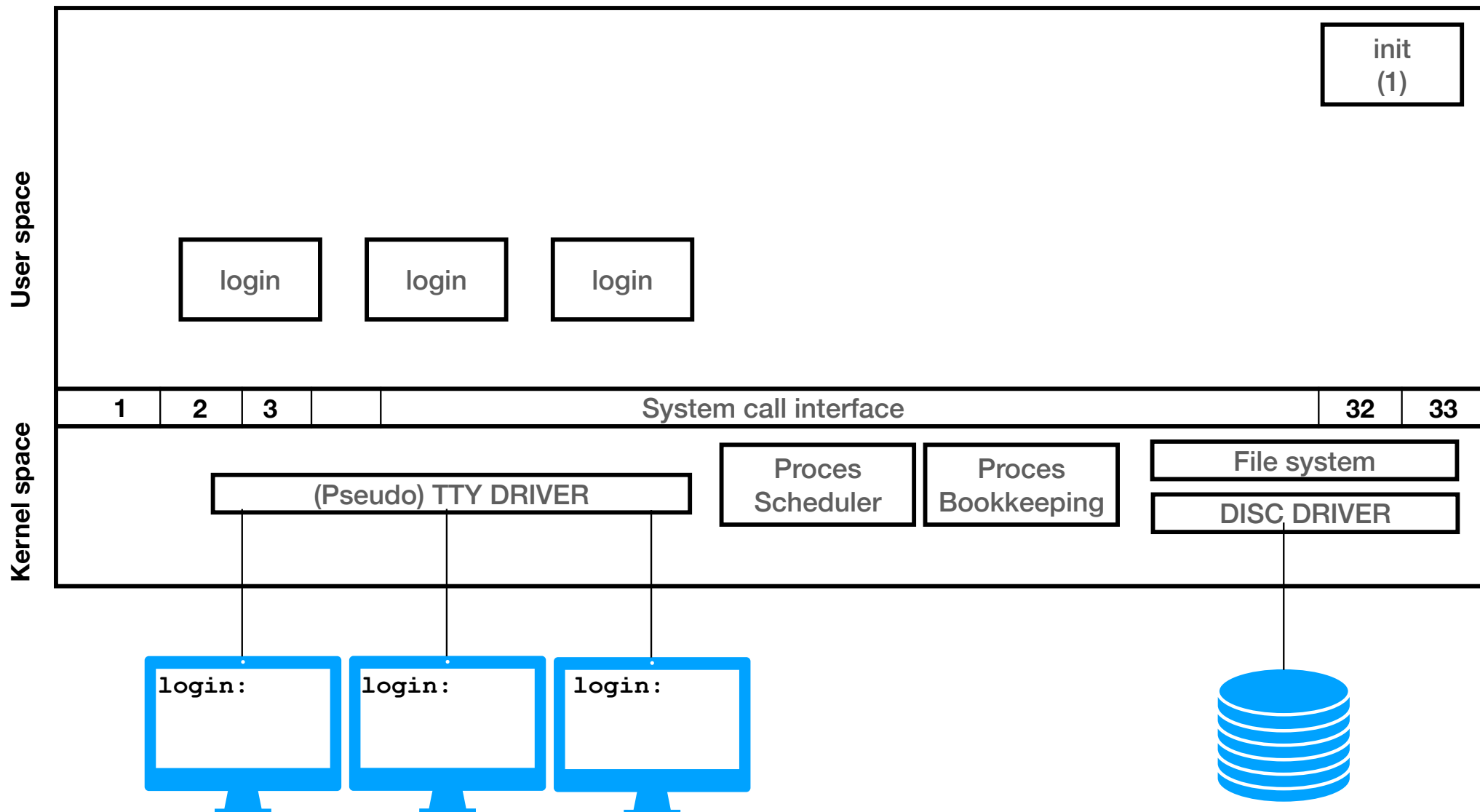
```
ls a* 2>errorfile | wc -l
0
```

0   `fork()`

**sh
pid 78**

1

---

0   **0**   `exec("ls")`

**sh   sh
pid 78   pid 89**

1   **1**

parent   child

---

0   `wait()`   **0**   `exit()`

**sh   ls
pid 78   pid 89**

1   **1**

0

**sh
pid 78**

1
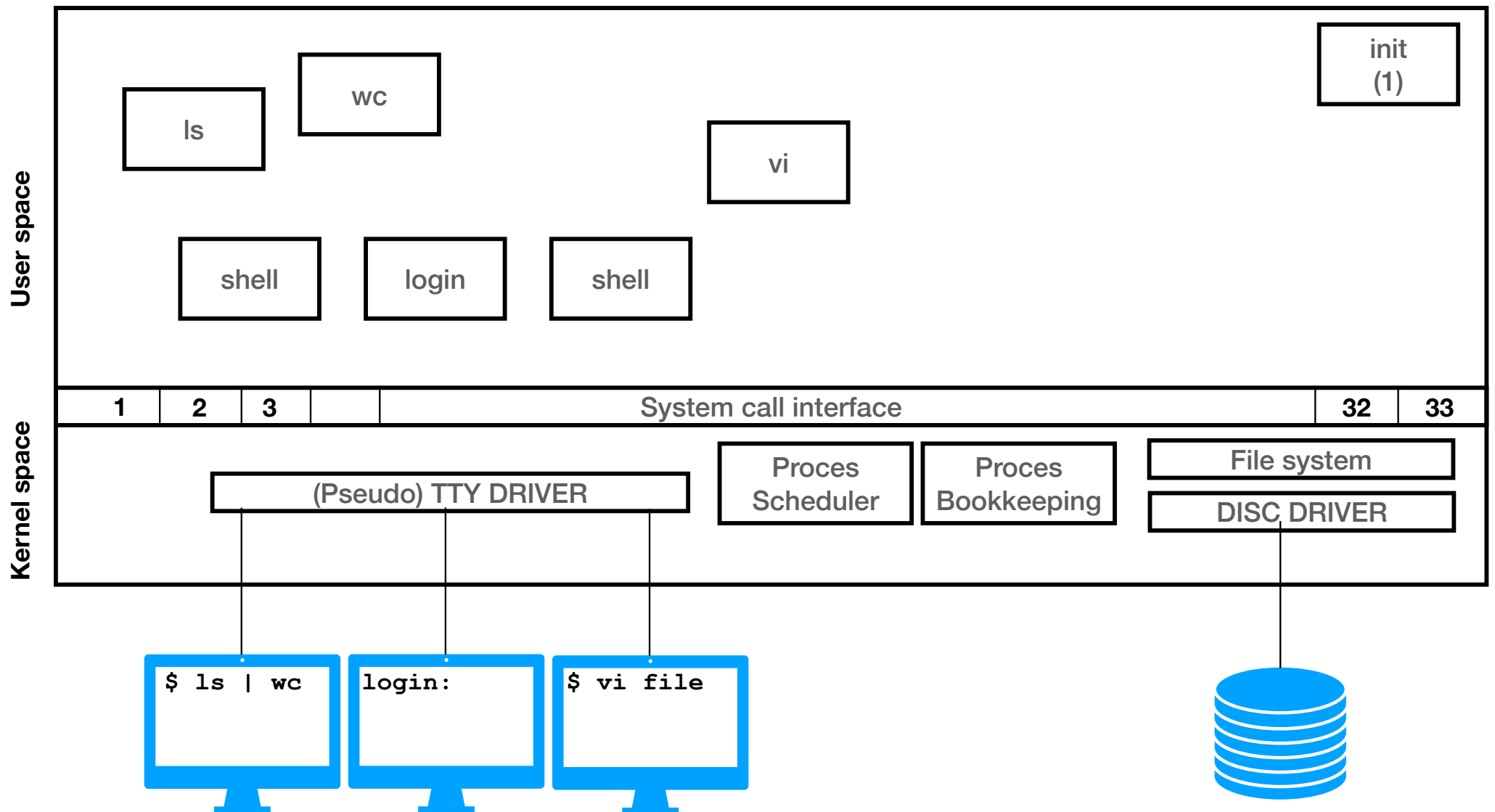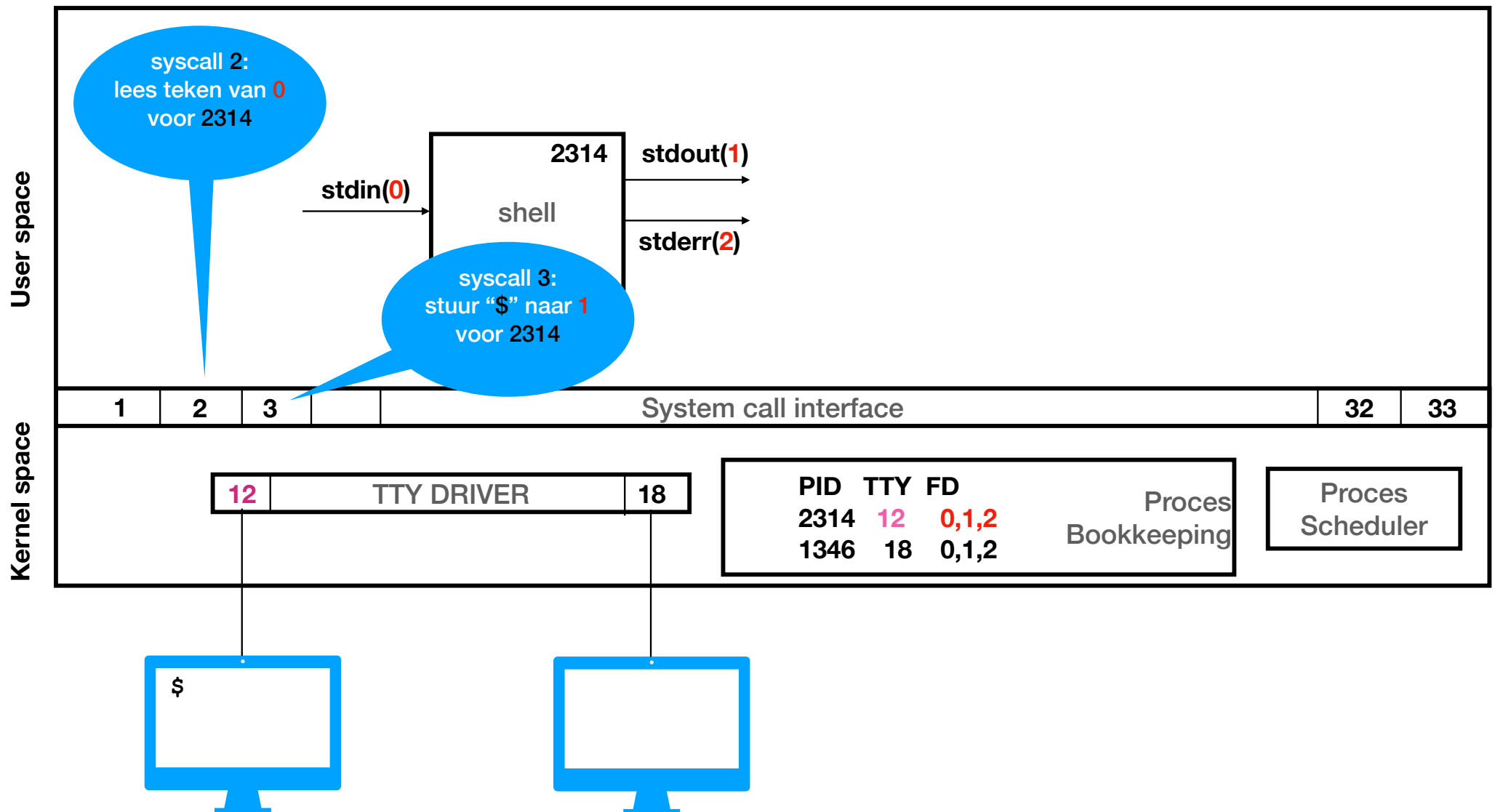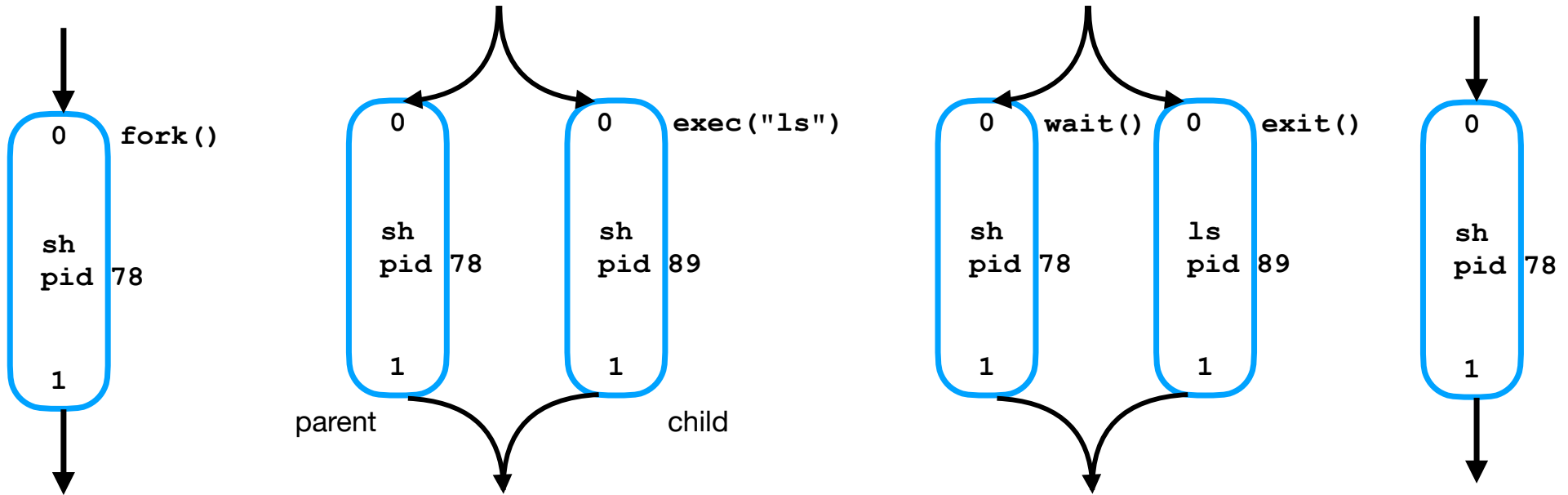
```
main() {

    int pid = fork();

    if (pid != 0){
       wait(pid)    // parent
    } else {
       exec("ls");  // child
    }

}
```

```
main() {

    int pid = fork();

    if (pid != 0){
       wait(pid)    // parent
    } else {
       exec("ls");  // child
    }

}
```

output redirection: `ls >outputfile`

```
main() {

    int fd;

    int pid = fork();

    if (pid != 0){
       wait(pid);      // parent
    } else {

      close(1);
      fd = creat("outputfile");   // fd wordt eerste
                                  // vrije file descriptor
                                  // en dat is dus 1

       exec("ls");   // child
    }

}
```

Een simpele shell

```
main() {
    int n, pid;
    char buffer[128];

    while (1) {

      write(1, "$ ", 2);

      n = read(0, buffer, 128);
      if (n == 0}{
         exit(0);
      }

      pid = fork();

      if (pid != 0){
       wait(pid);        // parent
      } else {
       exec(buffer);   // child
      }
    }
}
```