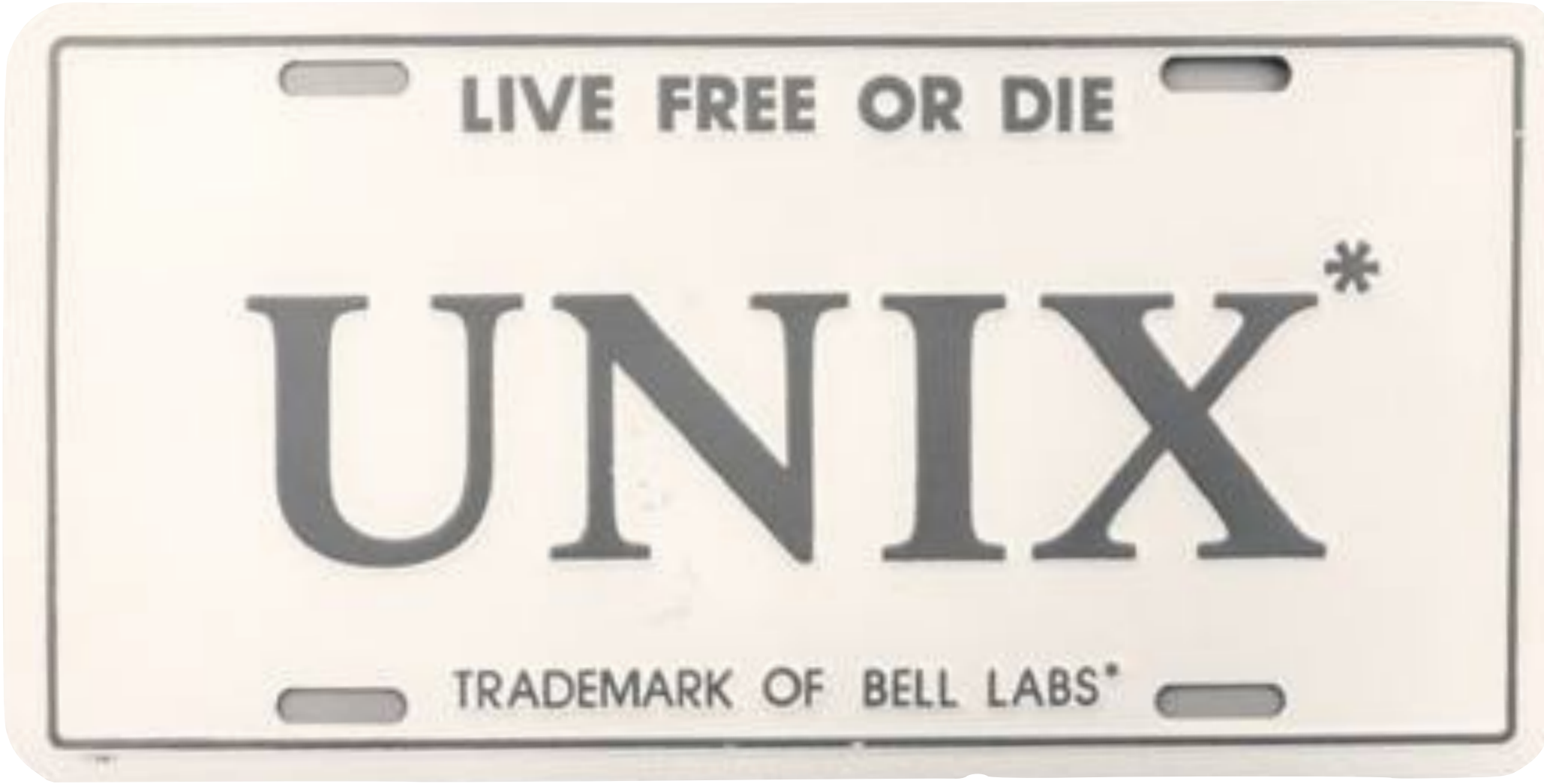

LIVE FREE OR DIE
UNIX*
TRADEMARK OF BELL LABS*

1969

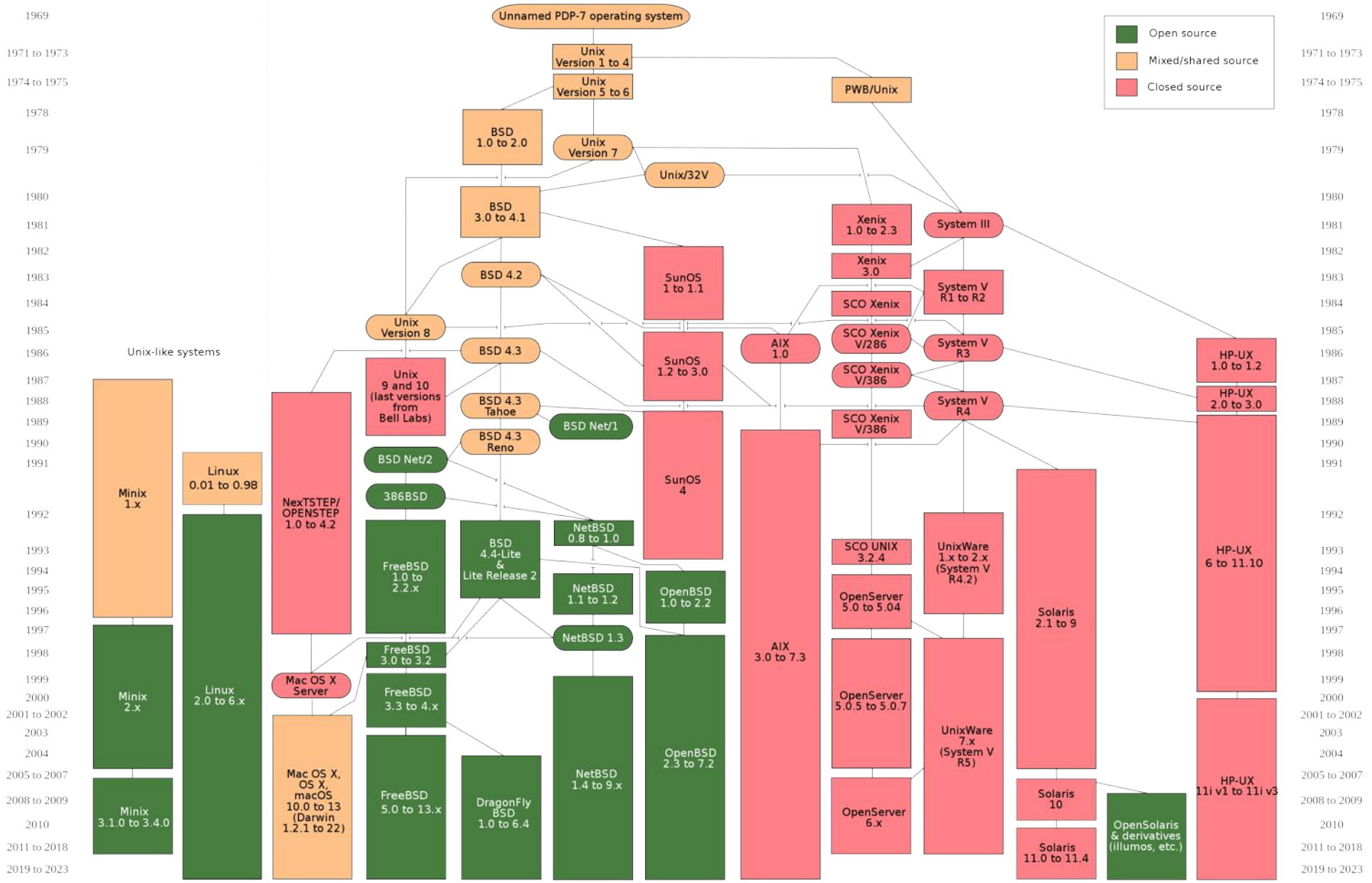
- American Telephone Company, AT&T:
  - ✴Telefooncentrales produceerden tabellen die met de hand verwerkt moesten worden
- Bell Labs (AT&T's research laboratorium):
  - ✴Produceerden nieuwe vindingen, 2 octrooi aanvragen per dag
  - Had de nieuwste DEC PDP 7 *mini computer* ($72.000, 500 KG)
    - Inspiratie 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

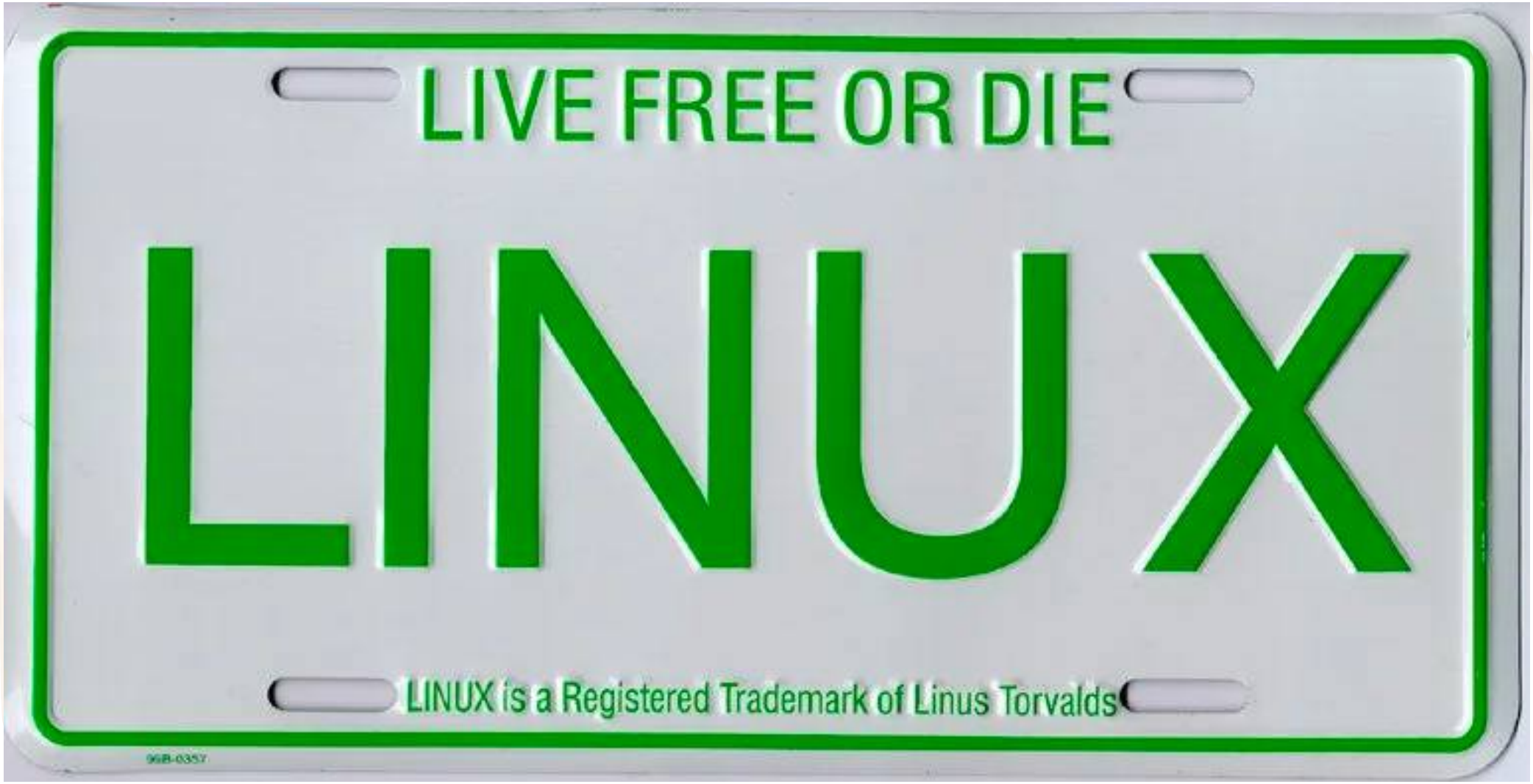
"...the number of UNIX installations has grown to 10, with more expected..."

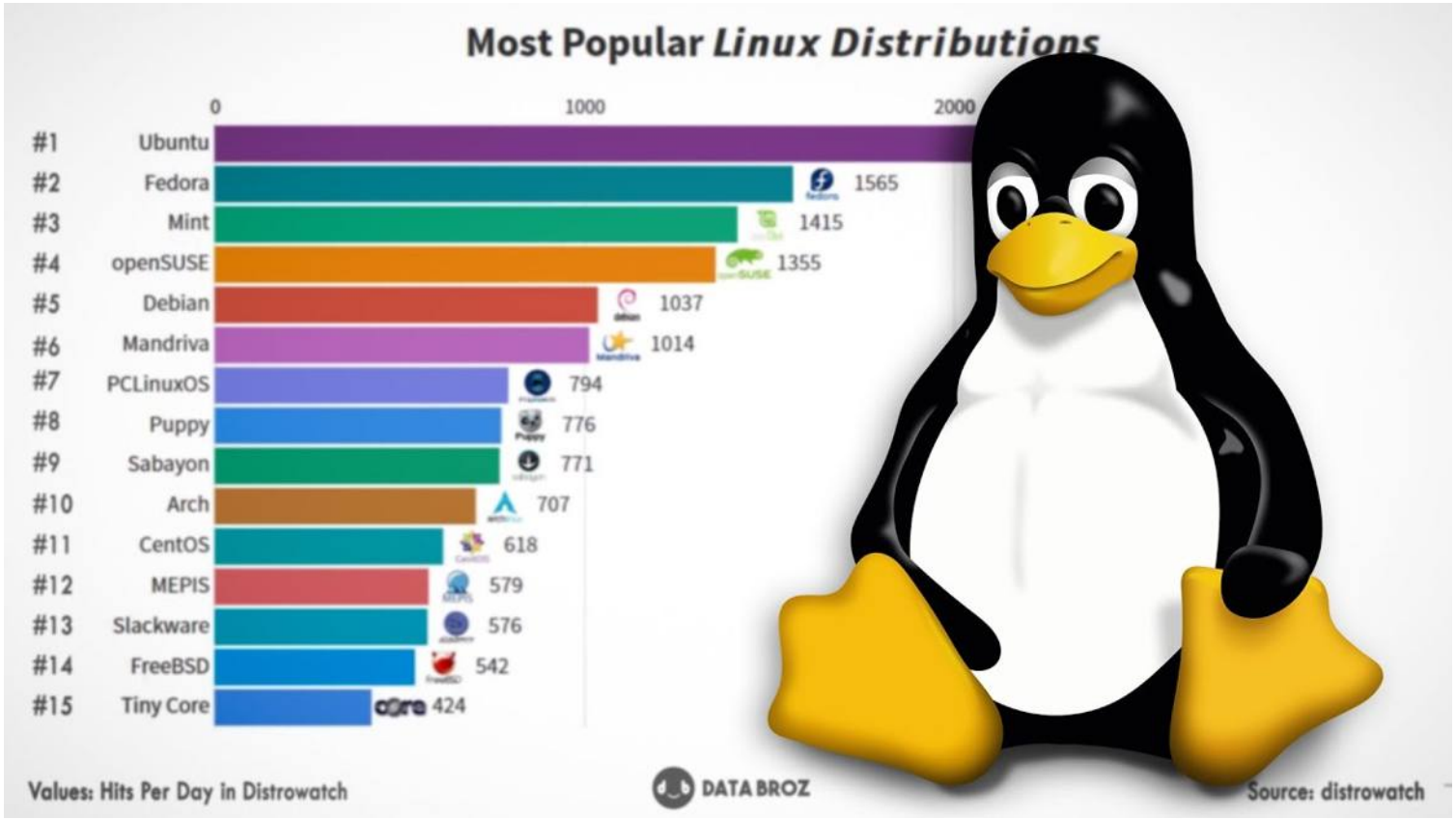
- Dennis Ritchie and Ken Thompson, June 1972

Leo Willems

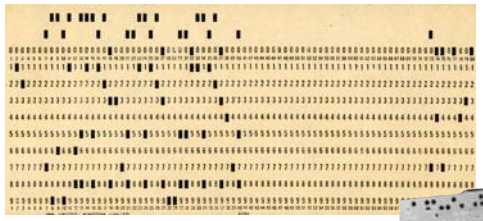
More history: https://www.youtube.com/watch?v=EY6q5dv_B-o


Unix-like systems
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
Open source
Mixed/shared source
Closed source
Unnamed PDP-7 operating system
Unix Version 1 to 4
Unix Version 5 to 6
PWB/Unix
BSD 1.0 to 2.0
Unix Version 7
Unix/32V
BSD 3.0 to 4.1
Xenix 1.0 to 2.3
System III
Xenix 3.0
BSD 4.2
SunOS 1 to 1.1
System V R1 to R2
SCO Xenix
Unix Version 8
SCO Xenix V/286
AIX 1.0
System V R3
HP-UX 1.0 to 1.2
BSD 4.3
SunOS 1.2 to 3.0
SCO Xenix V/386
Unix 9 and 10 (last versions from Bell Labs)
BSD 4.3 Tahoe
BSD Net/1
System V R4
HP-UX 2.0 to 3.0
BSD 4.3 Reno
SCO Xenix V/386
Minix 1.x
Linux 0.01 to 0.98
NexTSTEP/ OPENSTEP 1.0 to 4.2
BSD Net/2
SunOS 4
386BSD
NetBSD 0.8 to 1.0
AIX 3.0 to 7.3
SCO UNIX 3.2.4
UnixWare 1.x to 2.x (System V R4.2)
Solaris 2.1 to 9
HP-UX 6 to 11.10
FreeBSD 1.0 to 2.2.x
BSD 4.4-Lite & Lite Release 2
NetBSD 1.1 to 1.2
OpenBSD 1.0 to 2.2
OpenServer 5.0 to 5.04
Minix 2.x
Linux 2.0 to 6.x
NetBSD 1.3
FreeBSD 3.0 to 3.2
OpenBSD 2.3 to 7.2
OpenServer 5.0.5 to 5.0.7
UnixWare 7.x (System V R5)
Mac OS X Server
FreeBSD 3.3 to 4.x
Minix 3.1.0 to 3.4.0
Mac OS X, OS X, macOS 10.0 to 13 (Darwin 1.2.1 to 22)
FreeBSD 5.0 to 13.x
DragonFly BSD 1.0 to 6.4
NetBSD 1.4 to 9.x
OpenServer 6.x
Solaris 10
OpenSolaris & derivatives (illumos, etc.)
HP-UX 11i v1 to 11i v3
Solaris 11.0 to 11.4

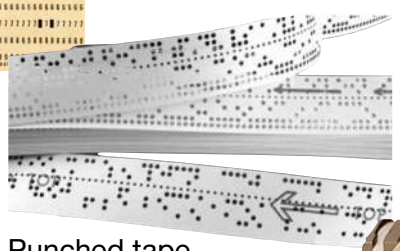

LIVE FREE OR DIE
LINUX
LINUX is a Registered Trademark of Linus Torvalds


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

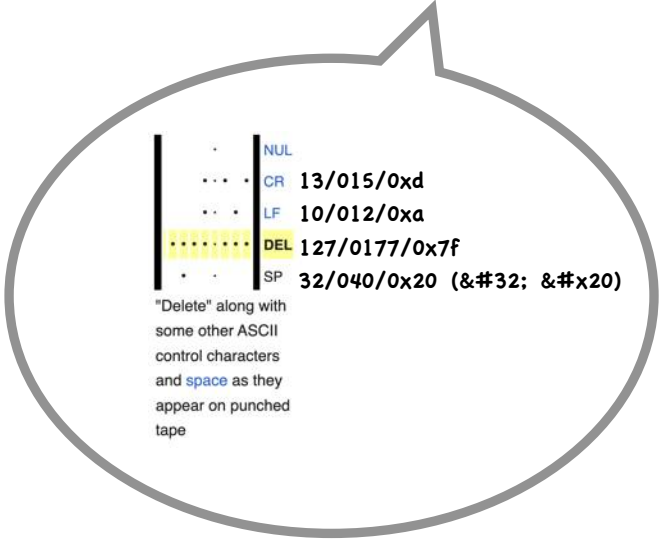
Input & Output



Ponskaart

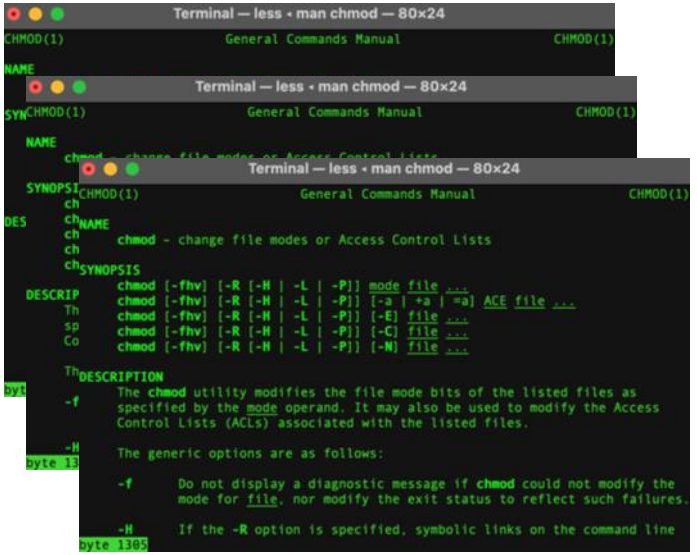


Punched tape



ca. 1957
Teletype (TTY)

Terminal (command line)
ADM3a



X11-Windows: Virtuele terminals

Directory aka 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

Options

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

soms is het een beetje vreemd

Input *of* Argumenten met de text utilities

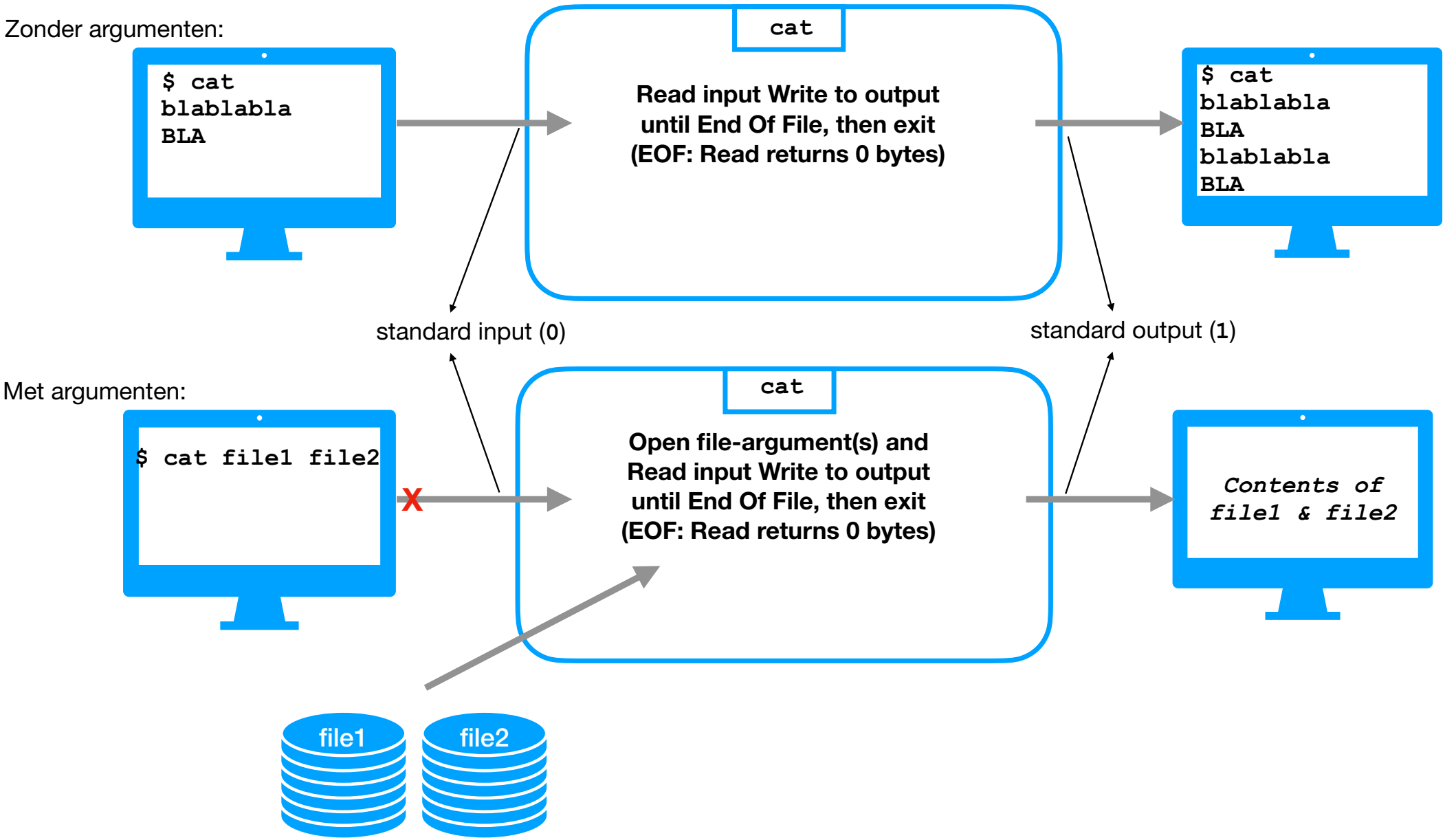
- **`cat`** zonder argumenten

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

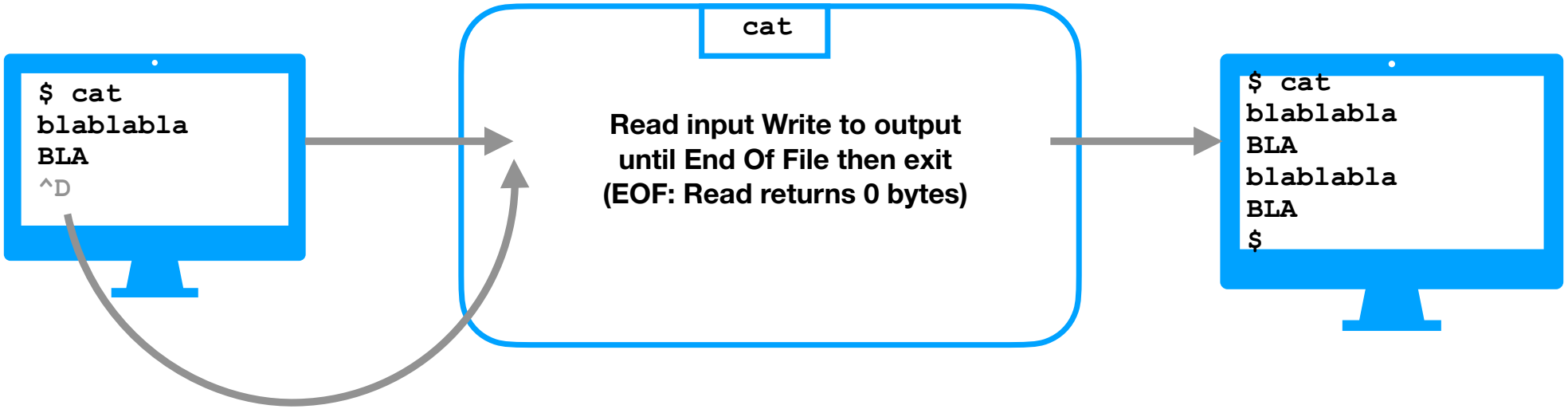
- **`cat file1 [file2 …]`**

  - leest de bestanden die worden opgegeven als argumenten

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

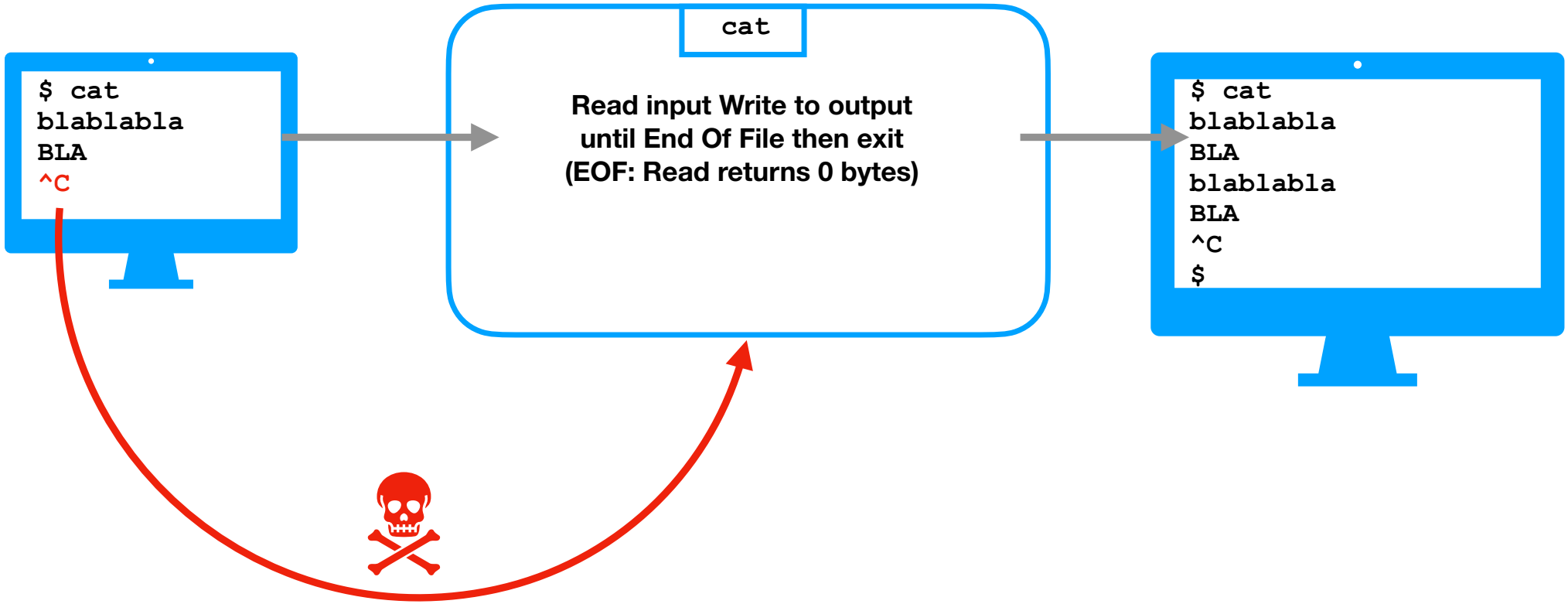

Zonder argumenten:
$ cat
blablabla
BLA
cat
Read input Write to output
until End Of File, then exit
(EOF: Read returns 0 bytes)
$ cat
blablabla
BLA
blablabla
BLA
standard input (0)
standard output (1)
Met argumenten:
$ cat file1 file2
X
cat
Open file-argument(s) and
Read input Write to output
until End Of File, then exit
(EOF: Read returns 0 bytes)
Contents of
file1 & file2
file1
file2

Input via de terminal stoppen: `^D`



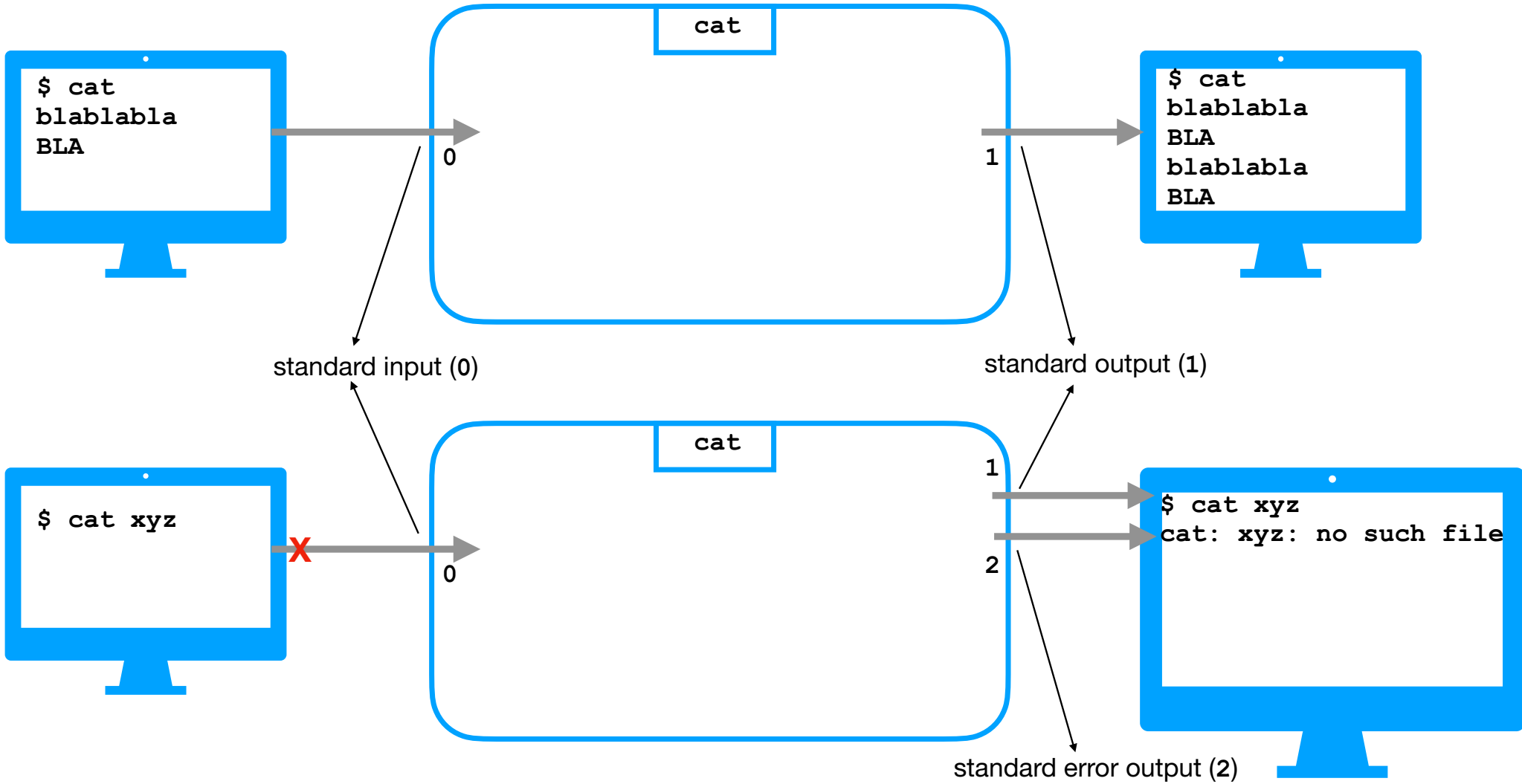
Kernel terminal device driver translates keyboard `^D` to: kernel sends 0 bytes (EOF)

Lopend programma afbreken via de terminal: `^C`



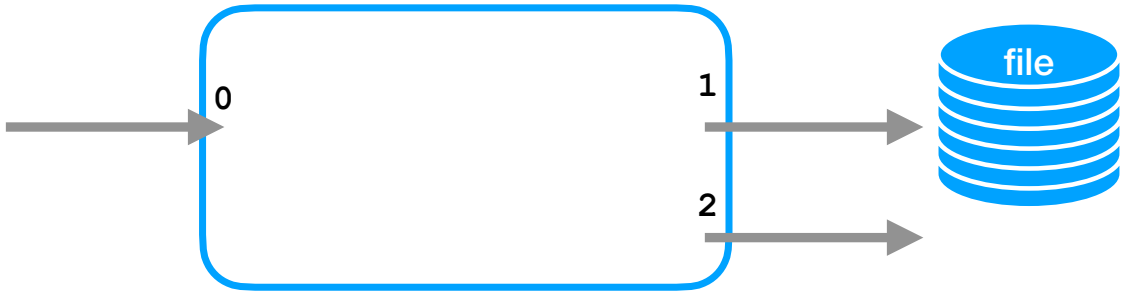
Kernel terminal device driver translates keyboard `^C` to: `kill -2` (terminal interrupt signal)

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

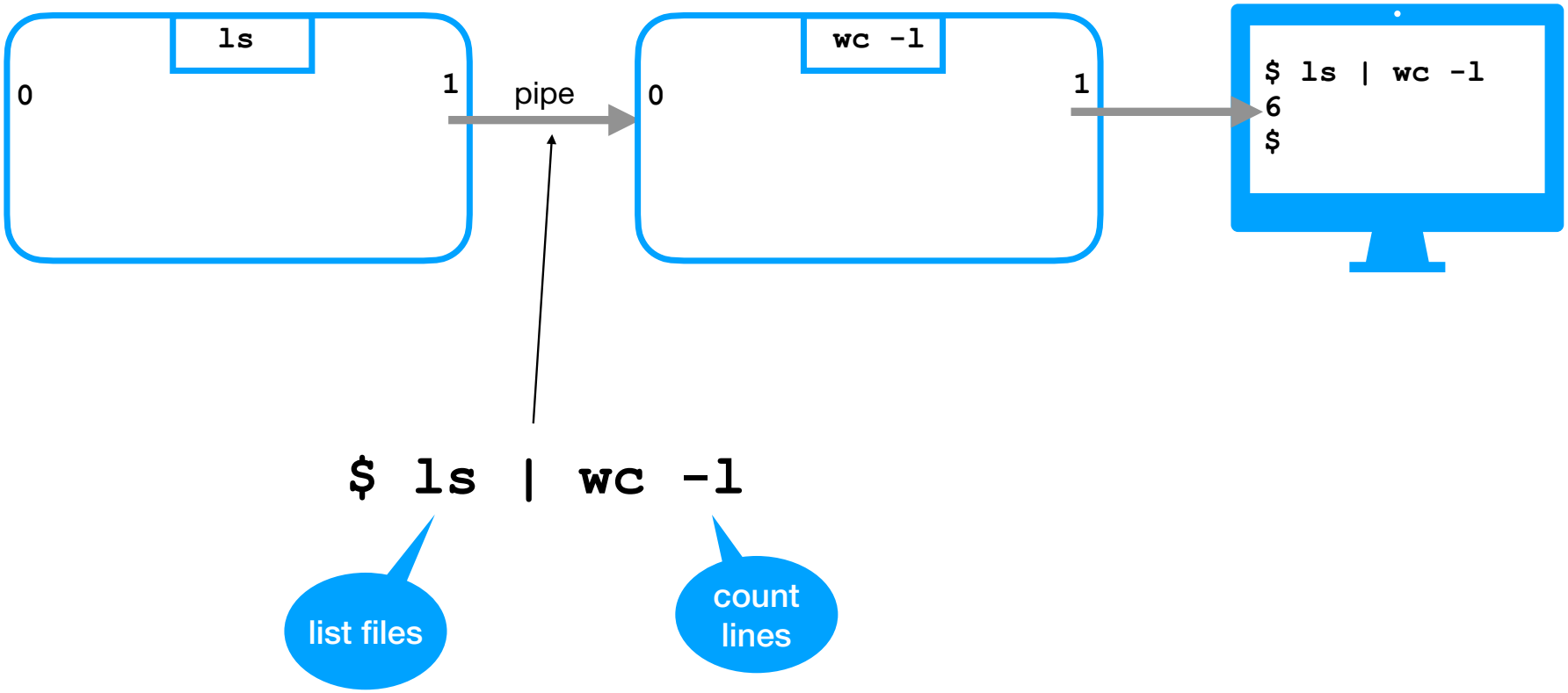
Combineer 2 met 1:

`$ cat 2>&1` close(2) and duplicate(1): merge 2 with 1

Append:

`$ cat >>output-file`

I/O Redirection: pipes



I/O Redirection

Weetje:
I/O redirection wordt gedaan door de shell, niet door de commando's zelf.
De shell koppelt 0,1 en en start dan pas commando('s)

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

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

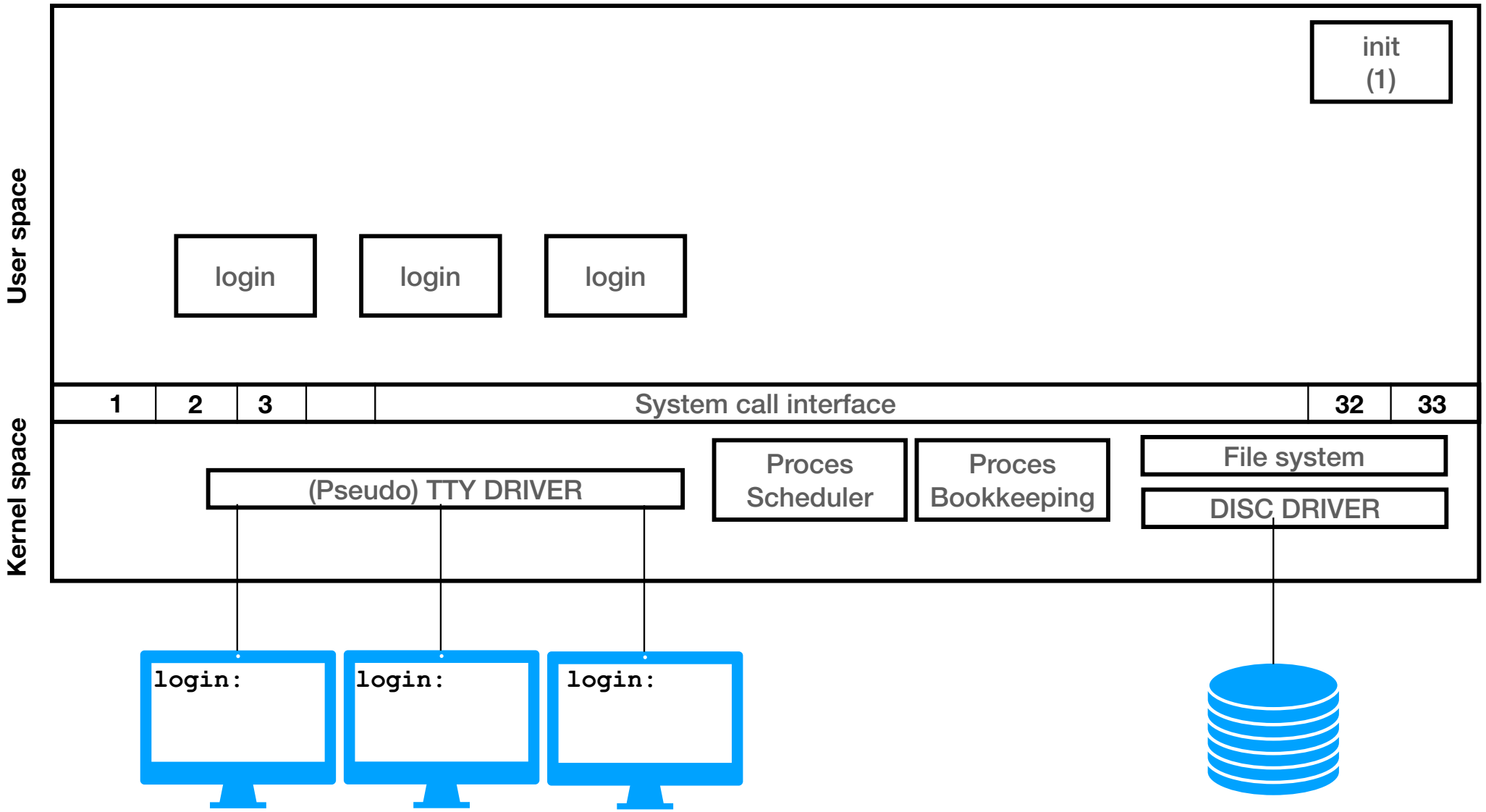
Stel dat er geen filenamen zijn die met een **a** beginnen, dan geeft **ls** een foutmelding:

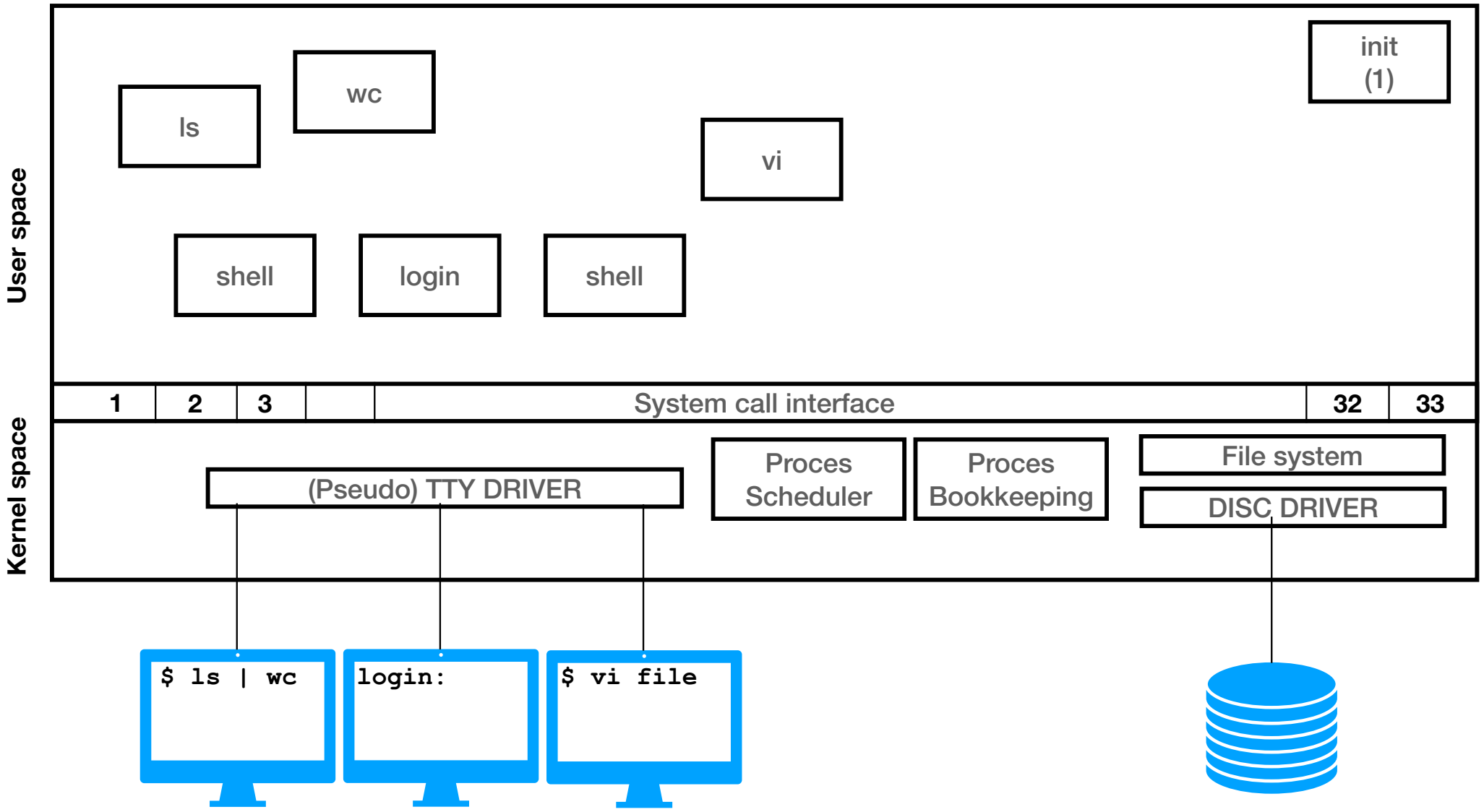
```
ls a* | wc -l
ls: a*: No such file or directory
0
```

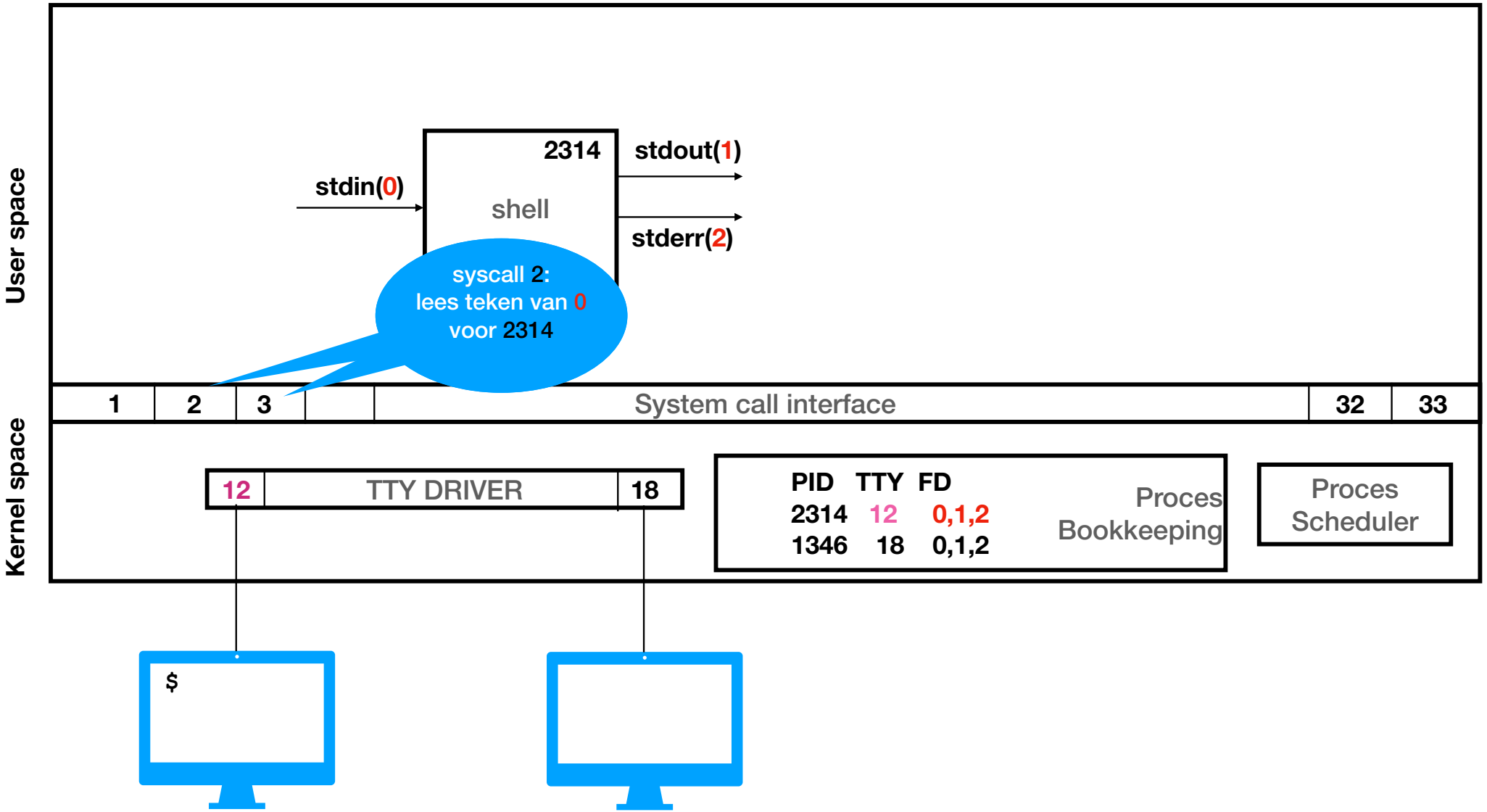
Weetje:
Als je geen <>| gebruikt dan "erft" een commando de 0,1 en 2 van de shell (en in de voorbeelden is dat de terminal)

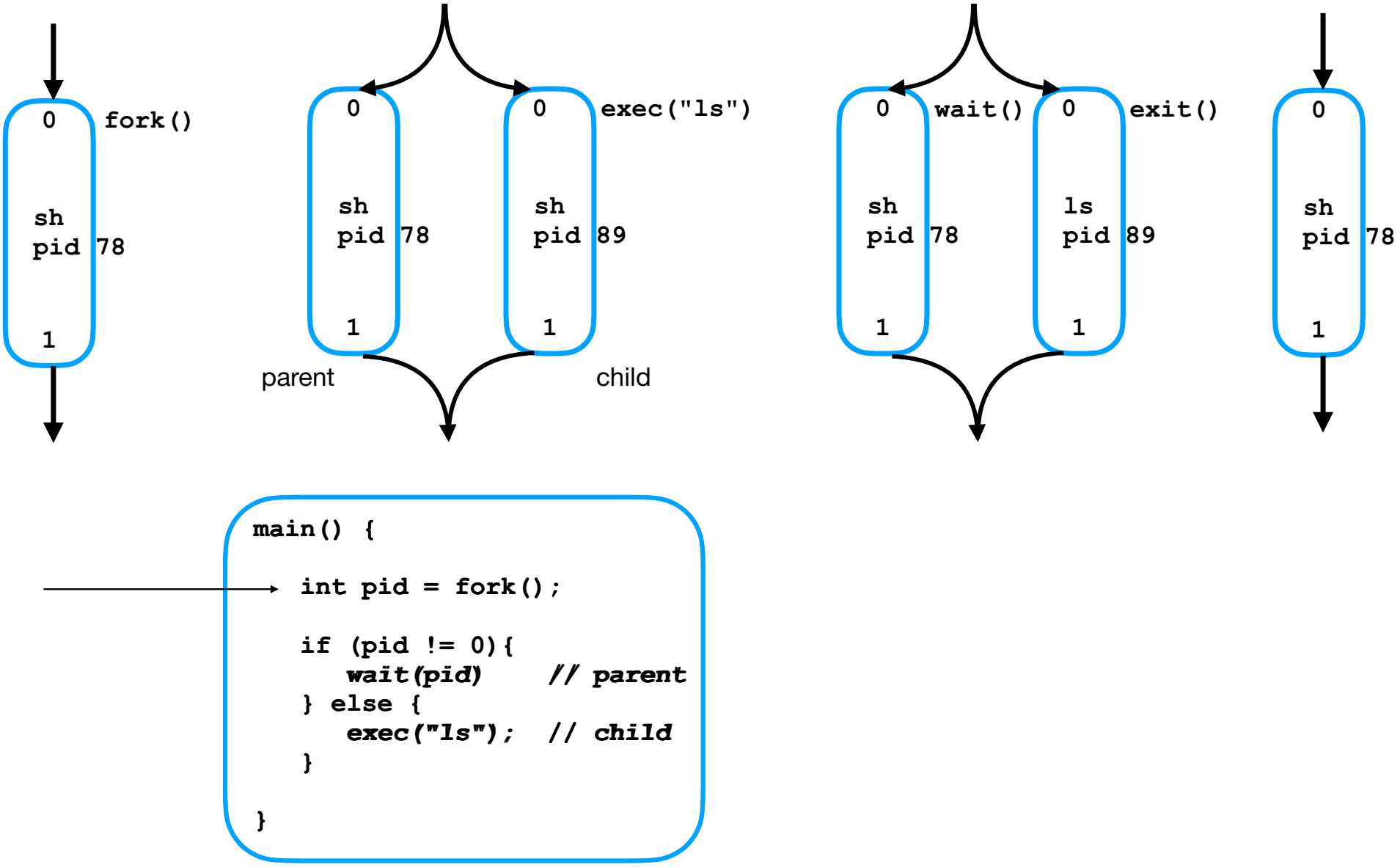
Onderdruk de foutmelding:

```
ls a* 2>errorfile | wc -l
0
```


User space
init
(1)
login
login
login
Kernel space
1
2
3
System call interface
32
33
(Pseudo) TTY DRIVER
Proces
Scheduler
Proces
Bookkeeping
File system
DISC DRIVER
login:
login:
login:


User space
ls
wc
vi
init
(1)
shell
login
shell
Kernel space
1
2
3
System call interface
32
33
(Pseudo) TTY DRIVER
Proces
Scheduler
Proces
Bookkeeping
File system
DISC DRIVER
$ ls | wc
login:
$ vi file


User space
Kernel space
2314
shell
stdin(0)
stdout(1)
stderr(2)
syscall 2:
lees teken van 0
voor 2314
1
2
3
System call interface
32
33
12
TTY DRIVER
18
PID TTY FD
2314 12 0,1,2
1346 18 0,1,2
Proces
Bookkeeping
Proces
Scheduler
$




```
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

     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
     }
  }
}
```