



FreeSoftware & Code Digging

LCM OpenLab

2019-05-10

Contenuti:

- Free Software:
 - Cos'è?
 - Perché è importante?
- Code Digging:
 - Concetti fondamentali:
 - Kernel
 - Syscalls
 - Processor basics
 - Tools:
 - Strace
 - Ltrace
 - GDB

FreeSoftware TEDx slides

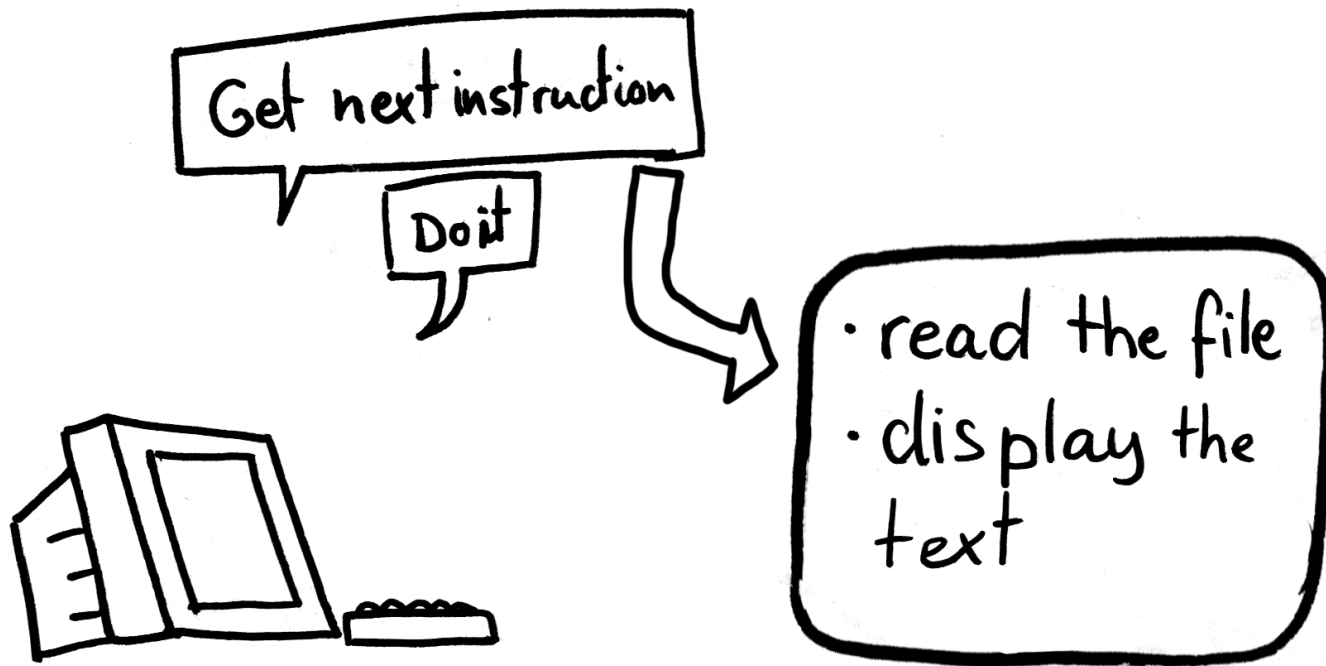
Copyright 2014 Christian Noguera,
Valentin Pasquier, Richard
Stallman

Released under CC-BY 3.0 license



LIBRE not GRATIS





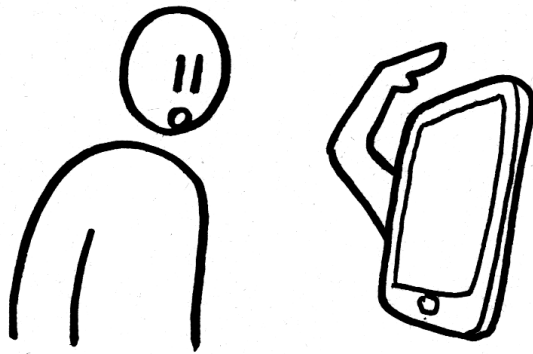


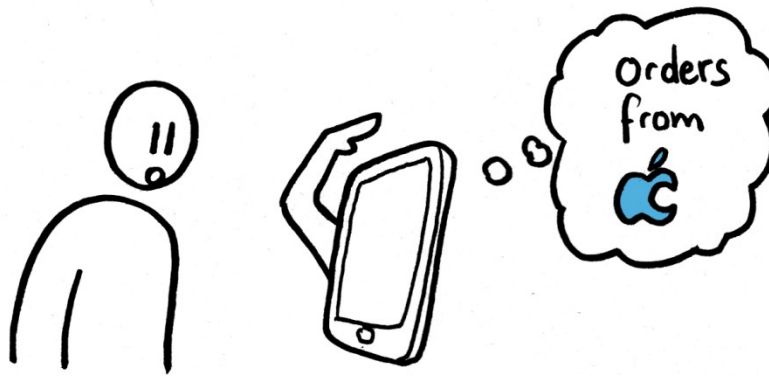
Who gives the instructions
to your computer ?



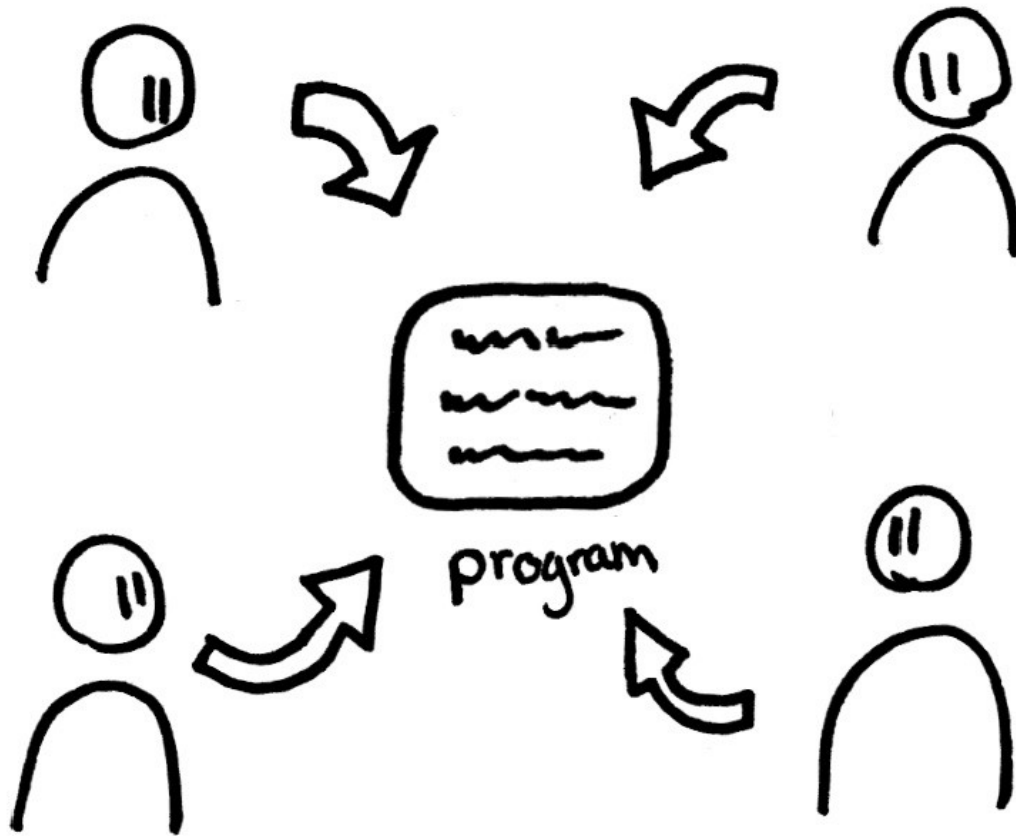




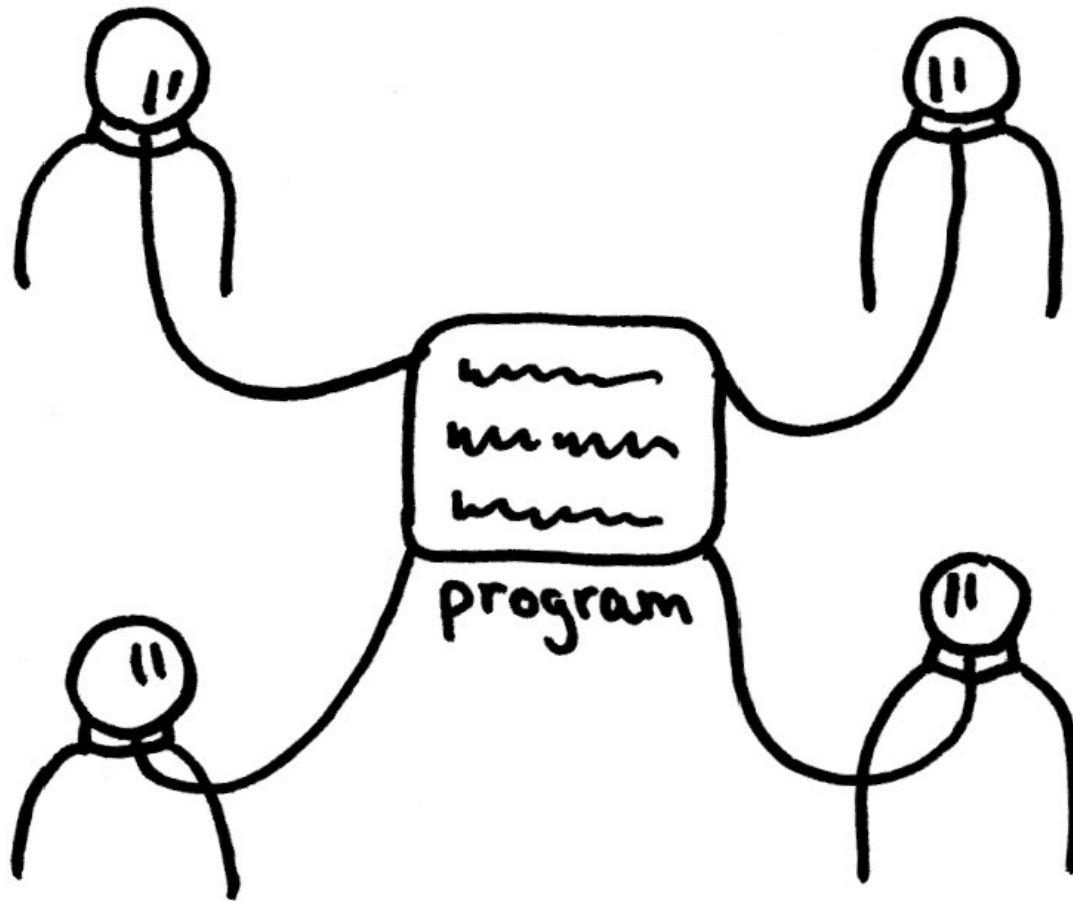




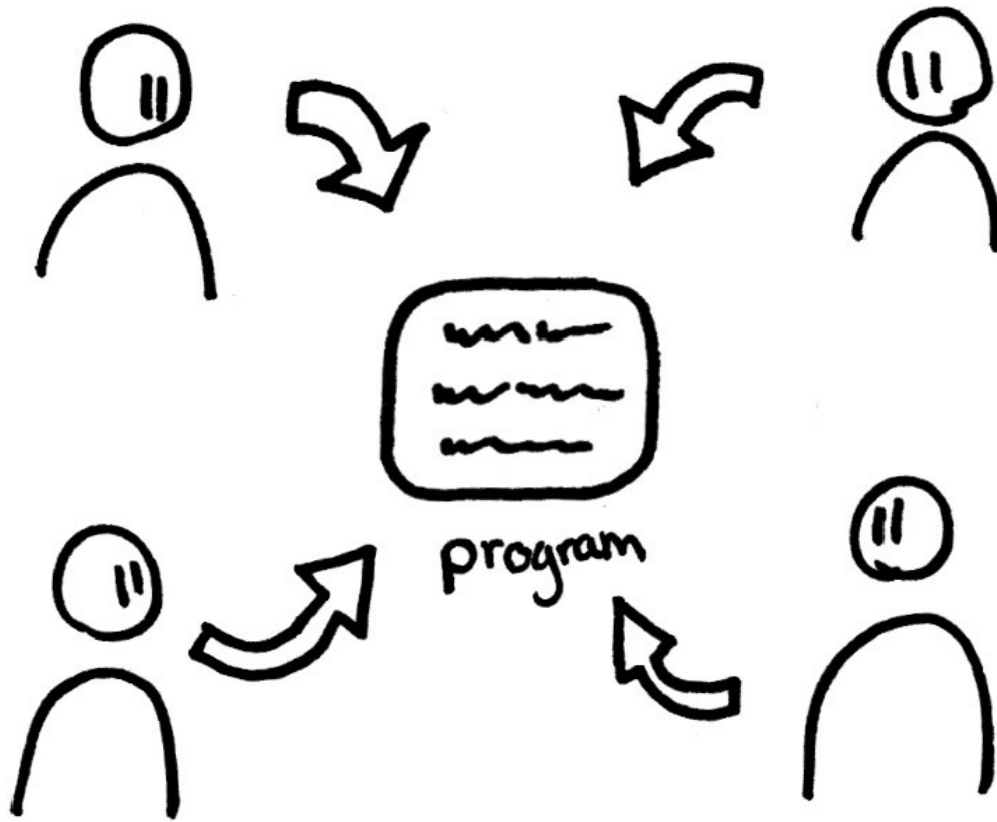
Users control the program



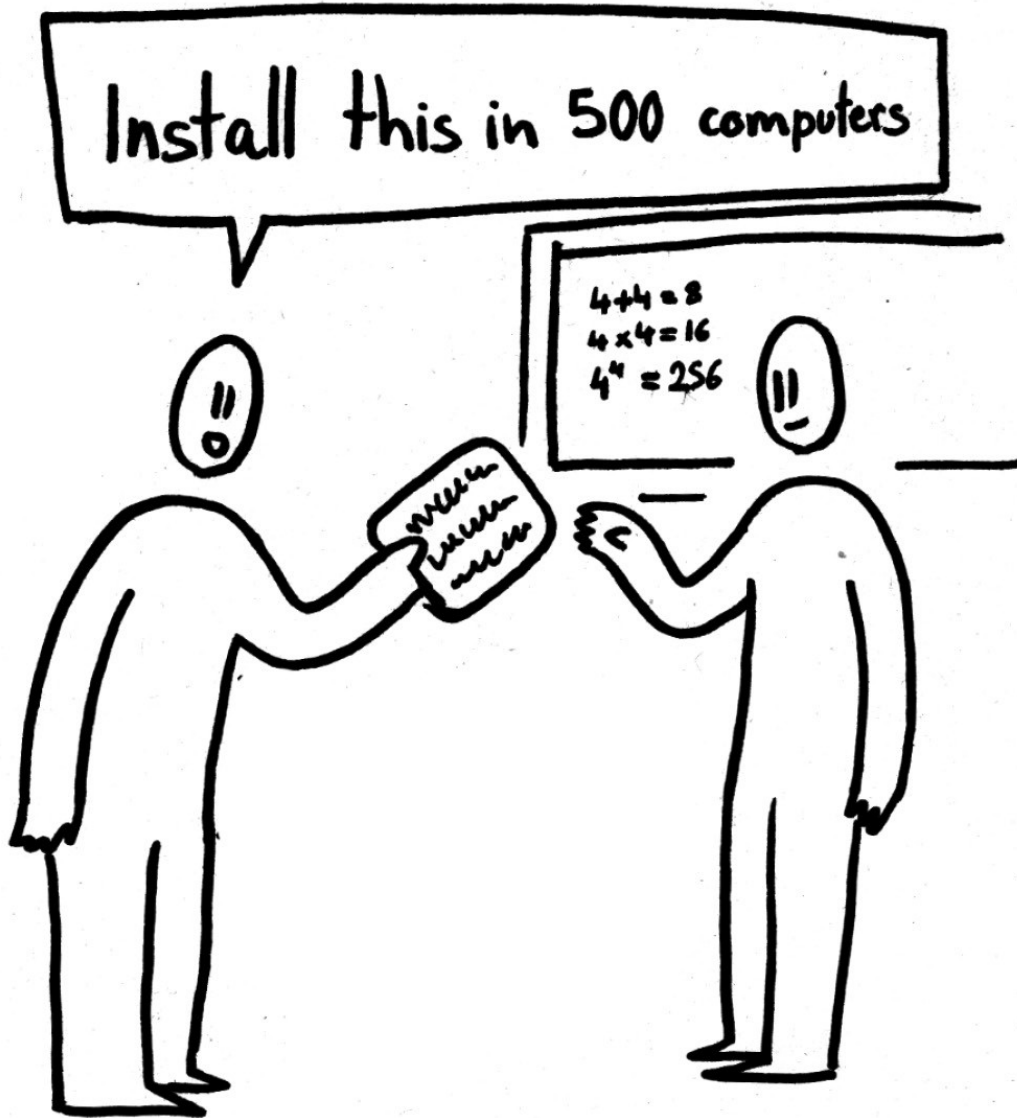
The program controls the users



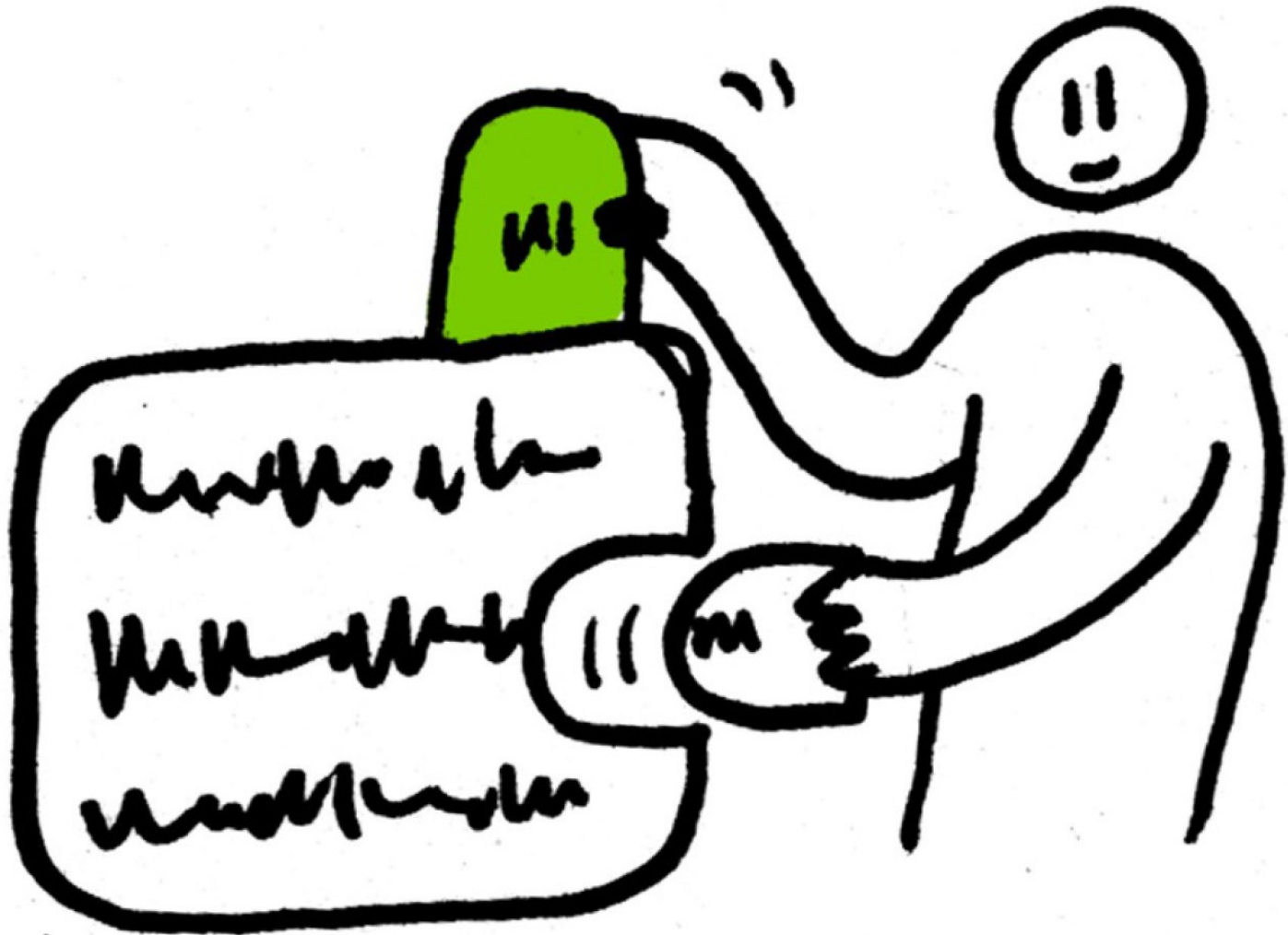
Freedoms 0, 1, 2, 3



Freedom 0
Run it



Freedom 1 – change it



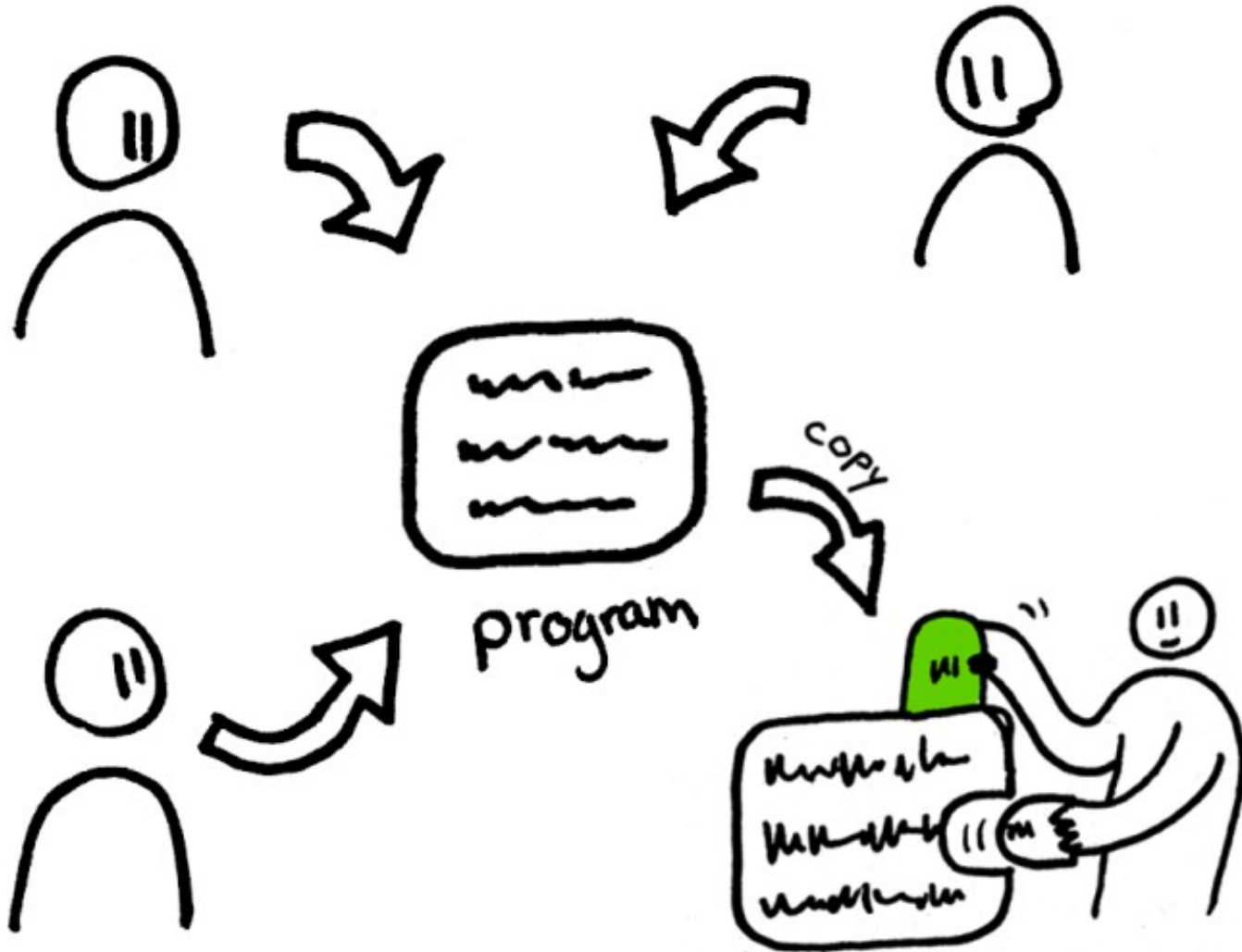

```
main()  
{  
printf("Hello,  
world!\n");  
}
```

Source

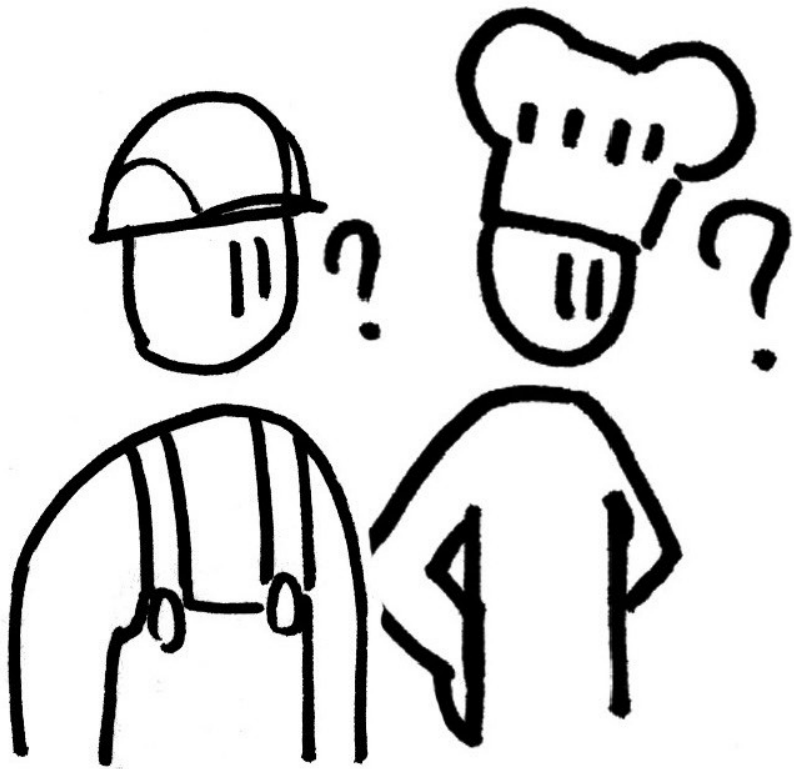
```
101101100100  
1001111011011  
0110011011101  
101110101011
```

Executable



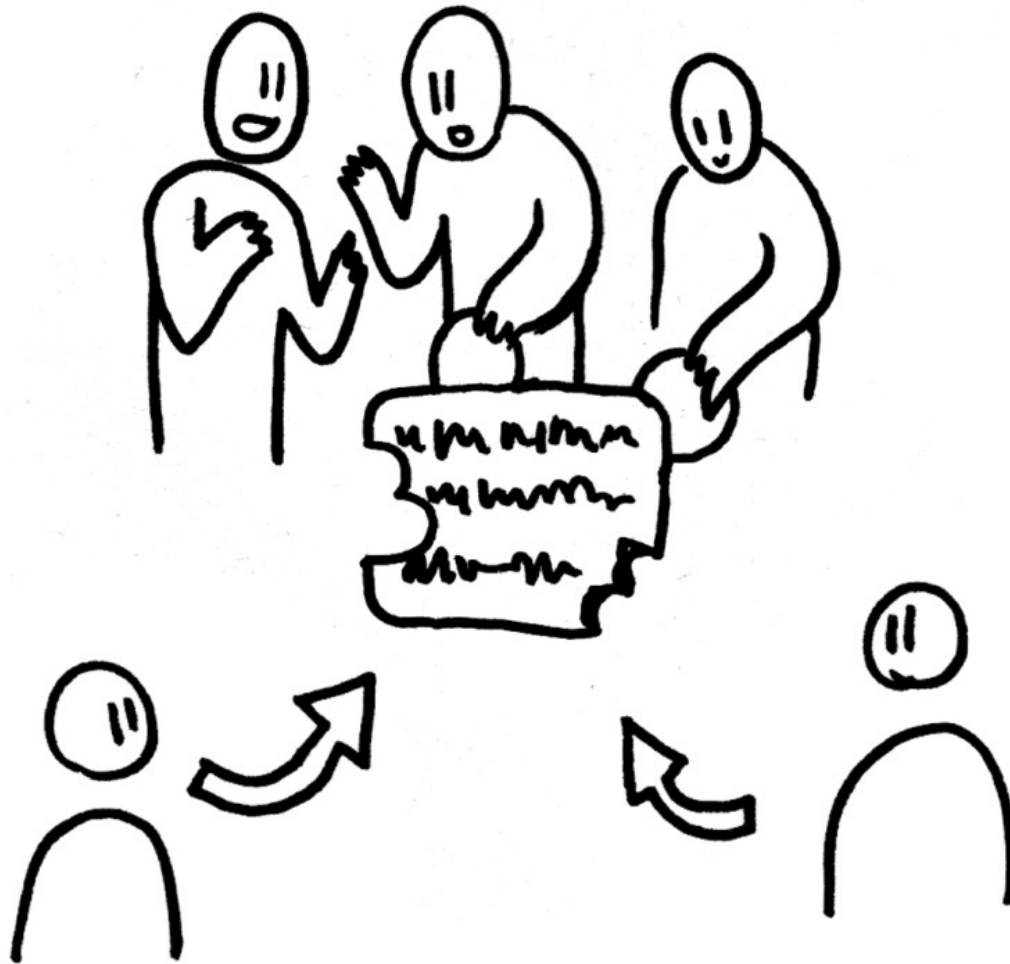


Not a programmer ?



```
int main(void)
{
    printf("Hello world")
}
```

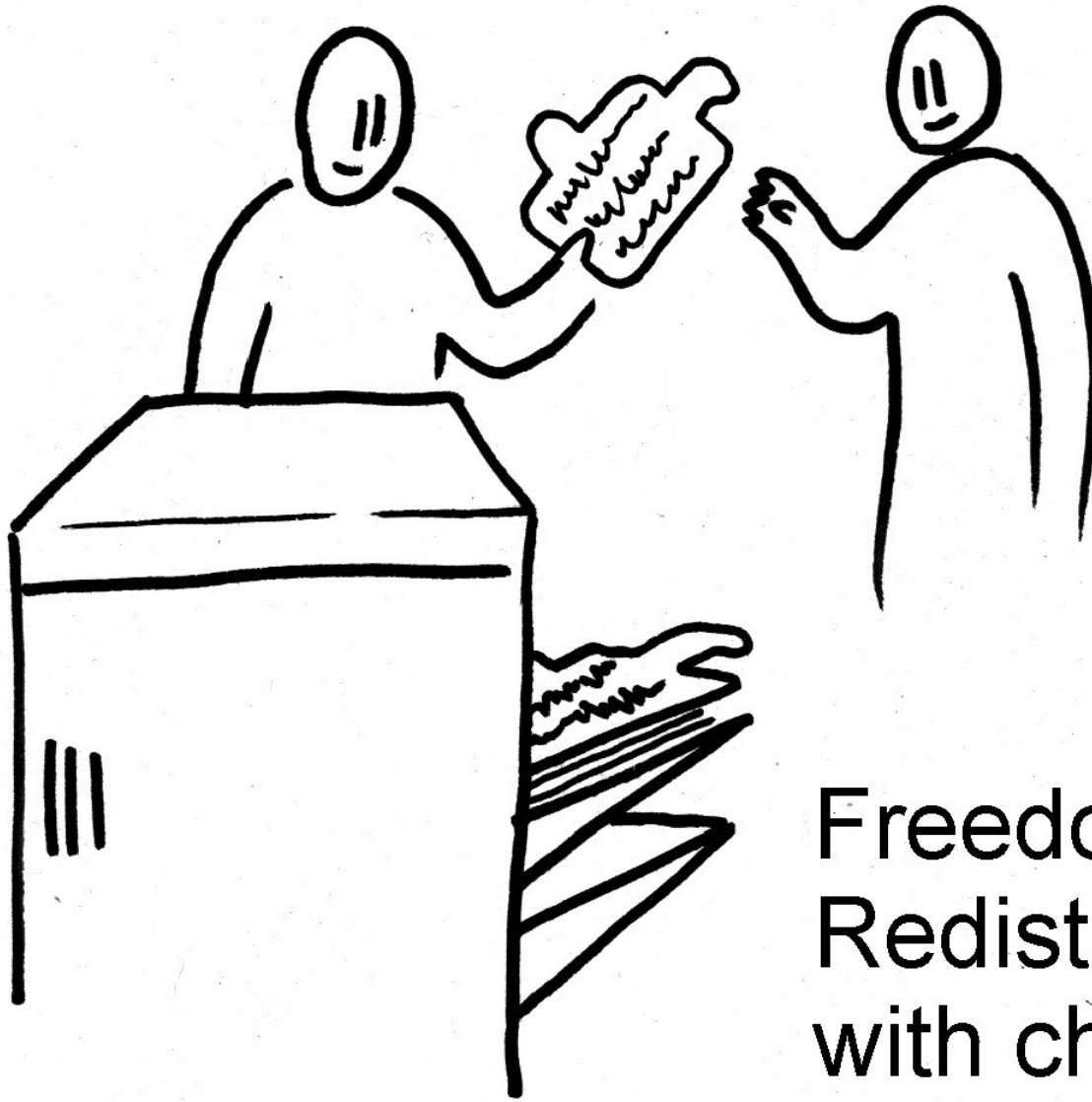






Freedom 2
Redistribute it

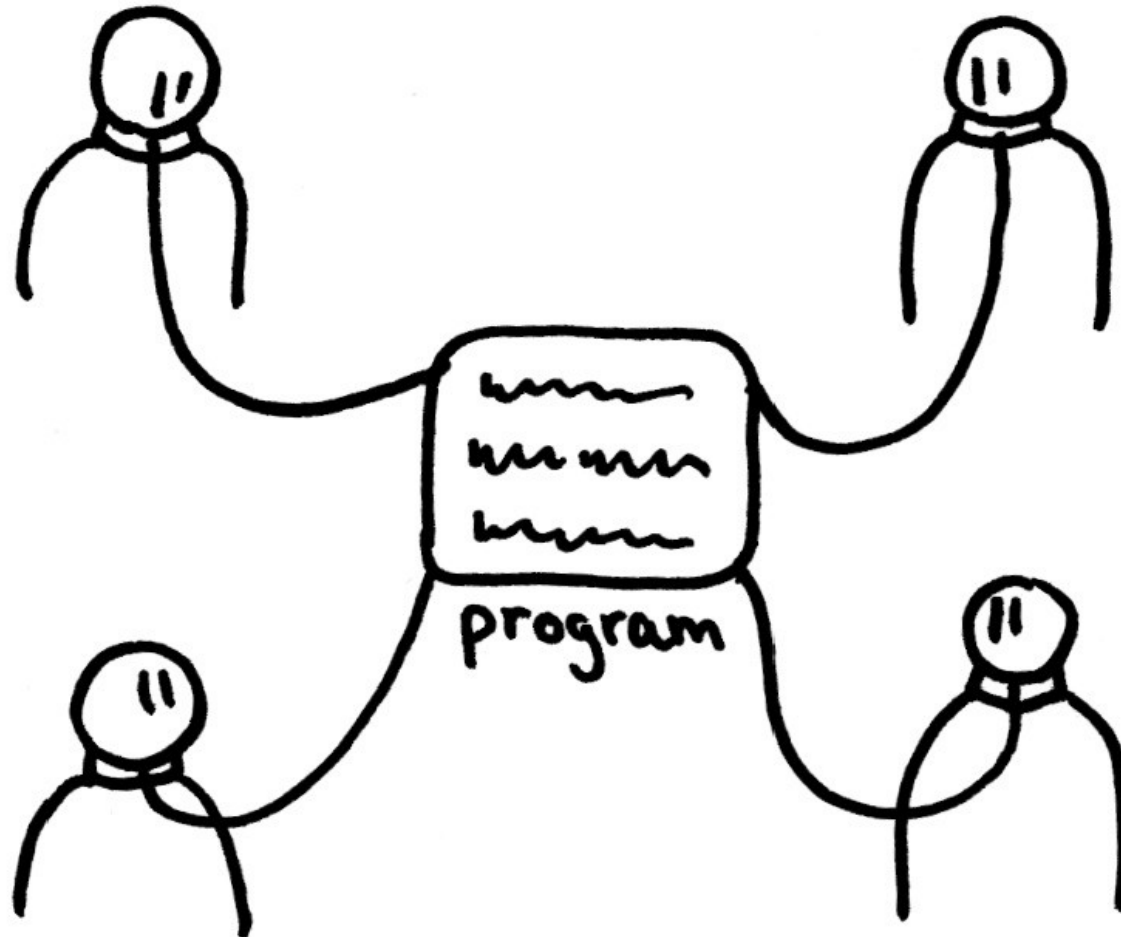


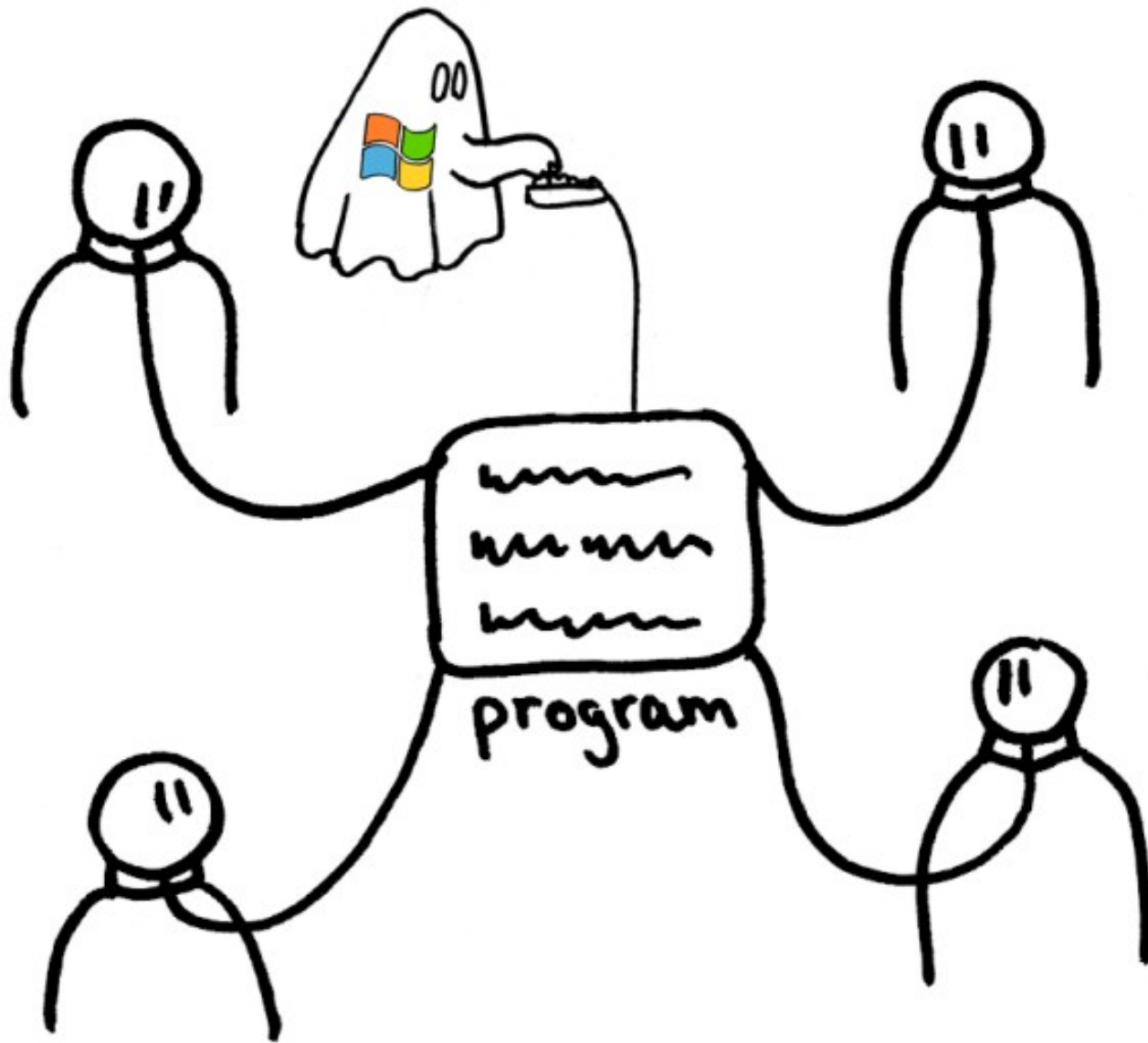


Freedom 3
Redistribute
with changes



The program controls the users

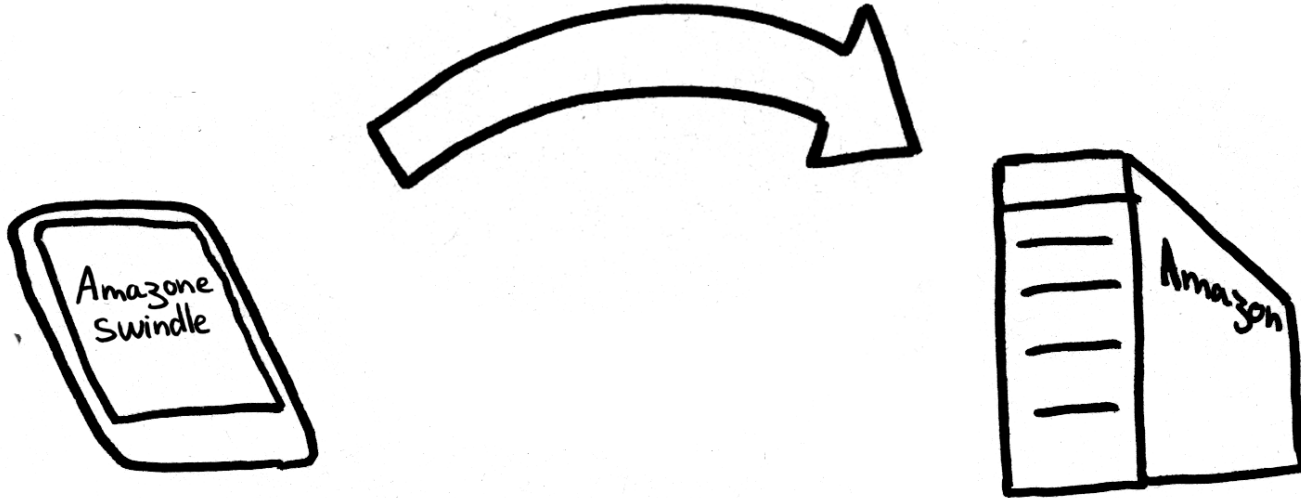


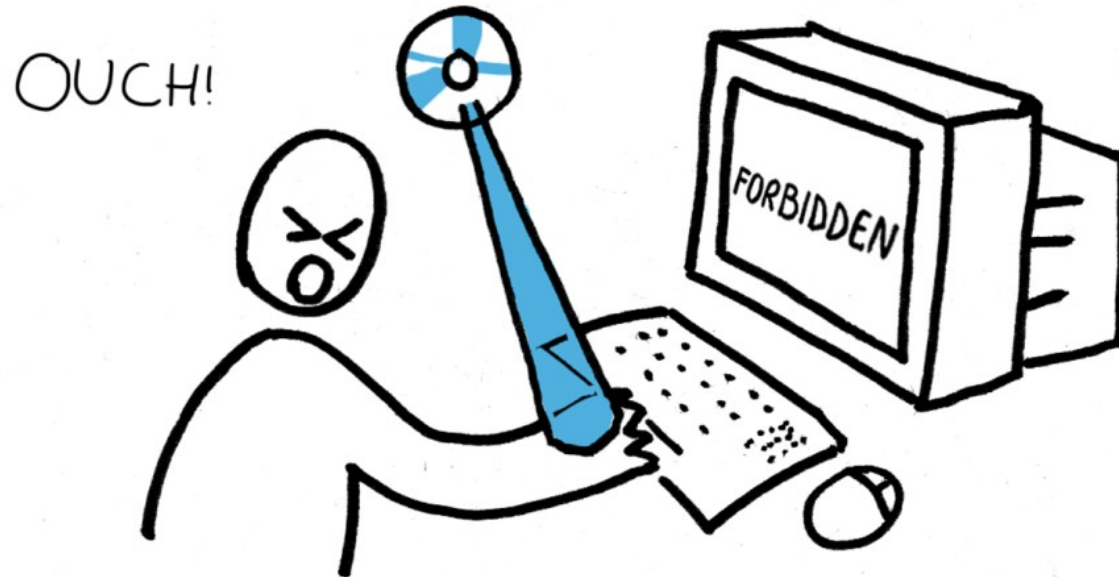


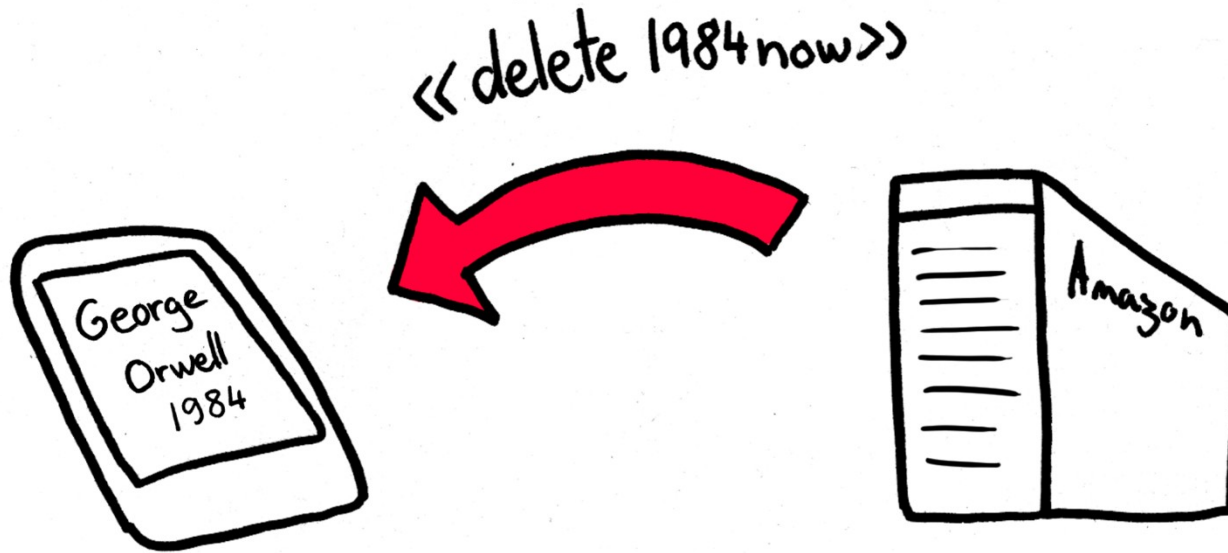
Shopper 8 is on isle 2!



« He's reading page 5 »









PS3



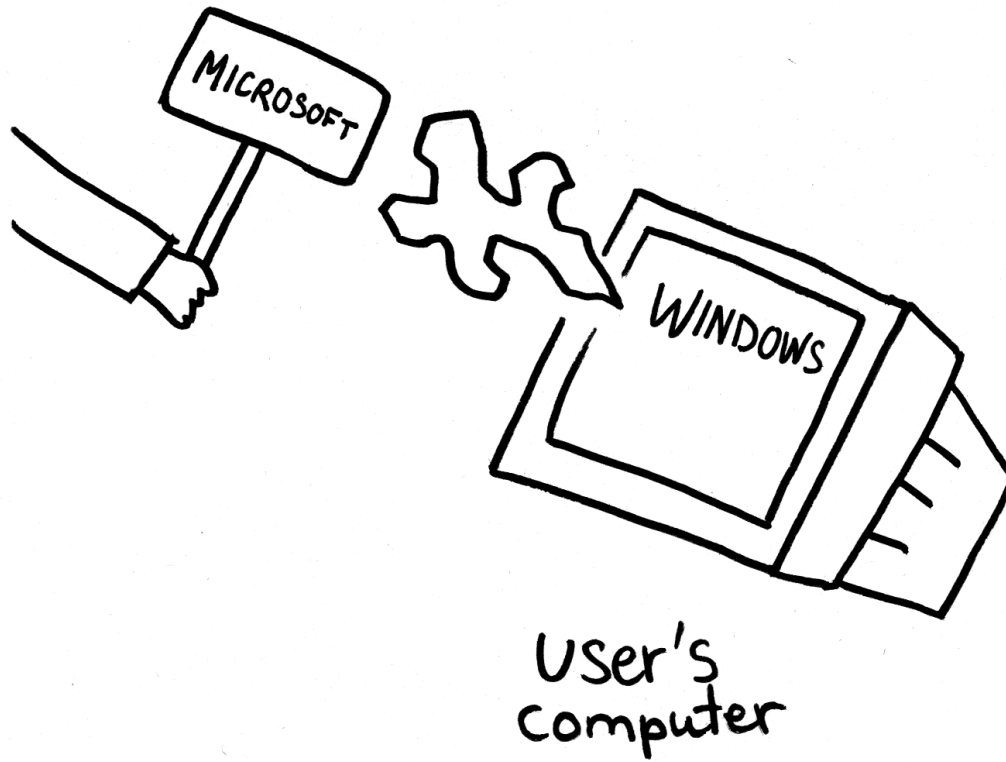
This will downgrade
your system. Proceed?

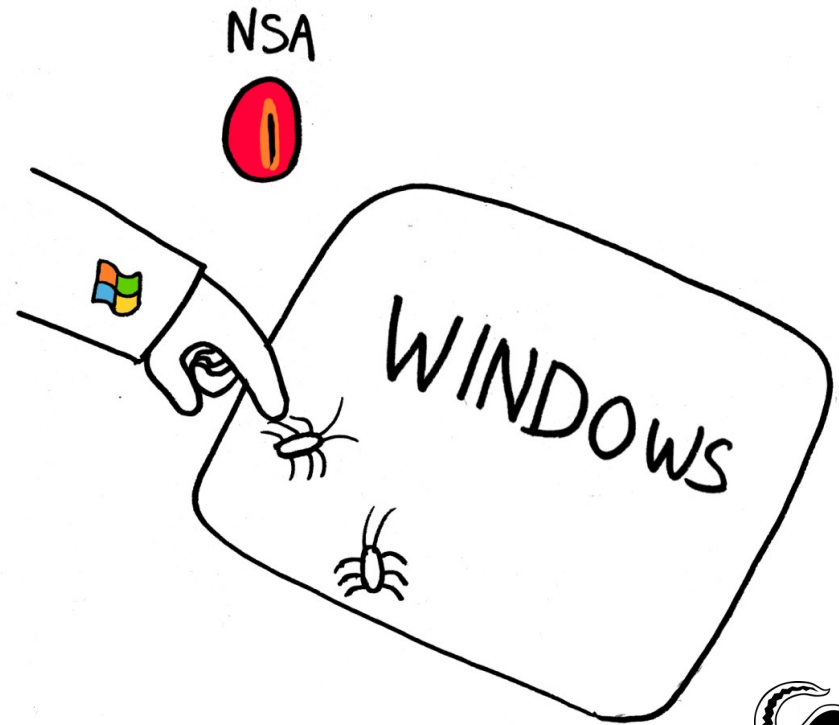
YES

NO*

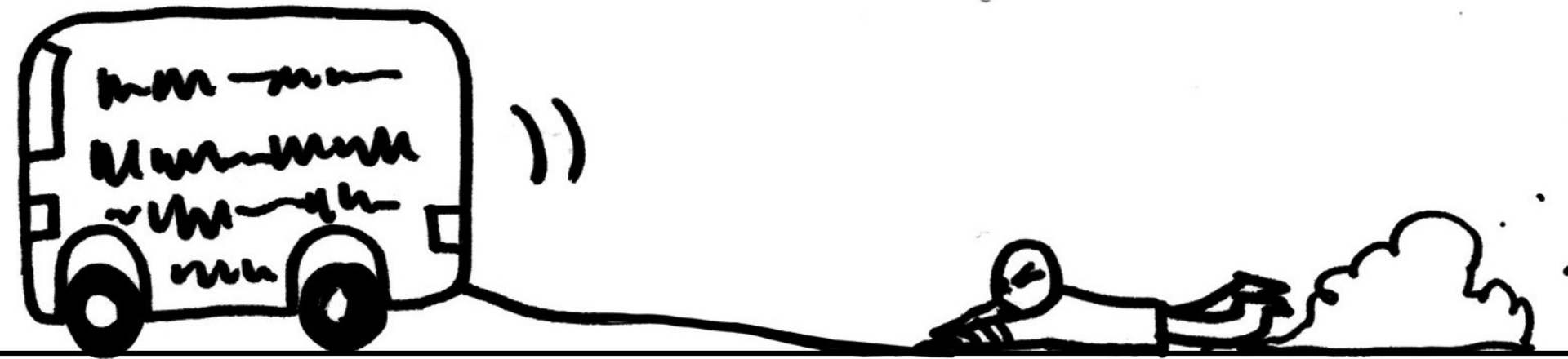
* Use on the net won't work anymore.

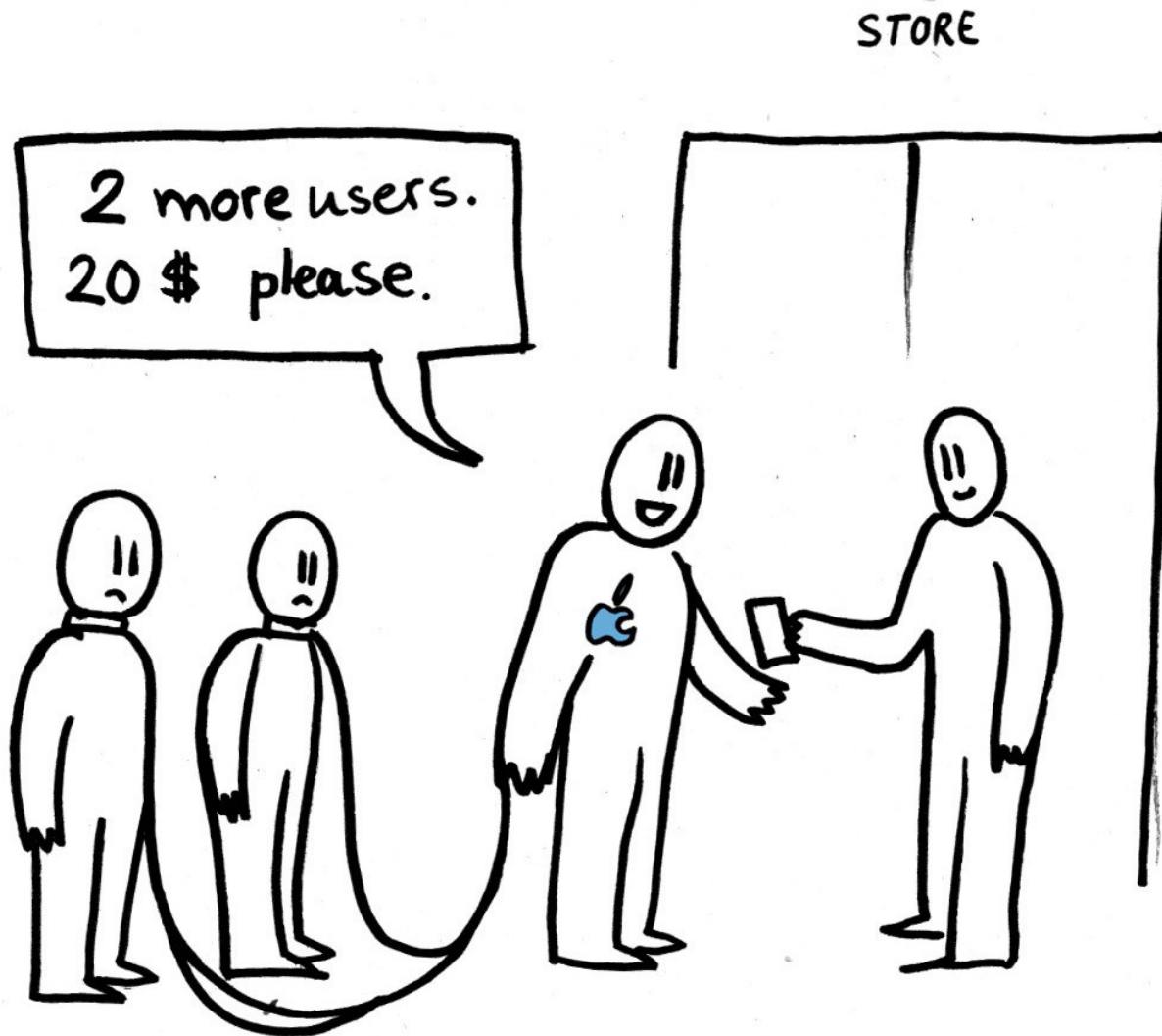






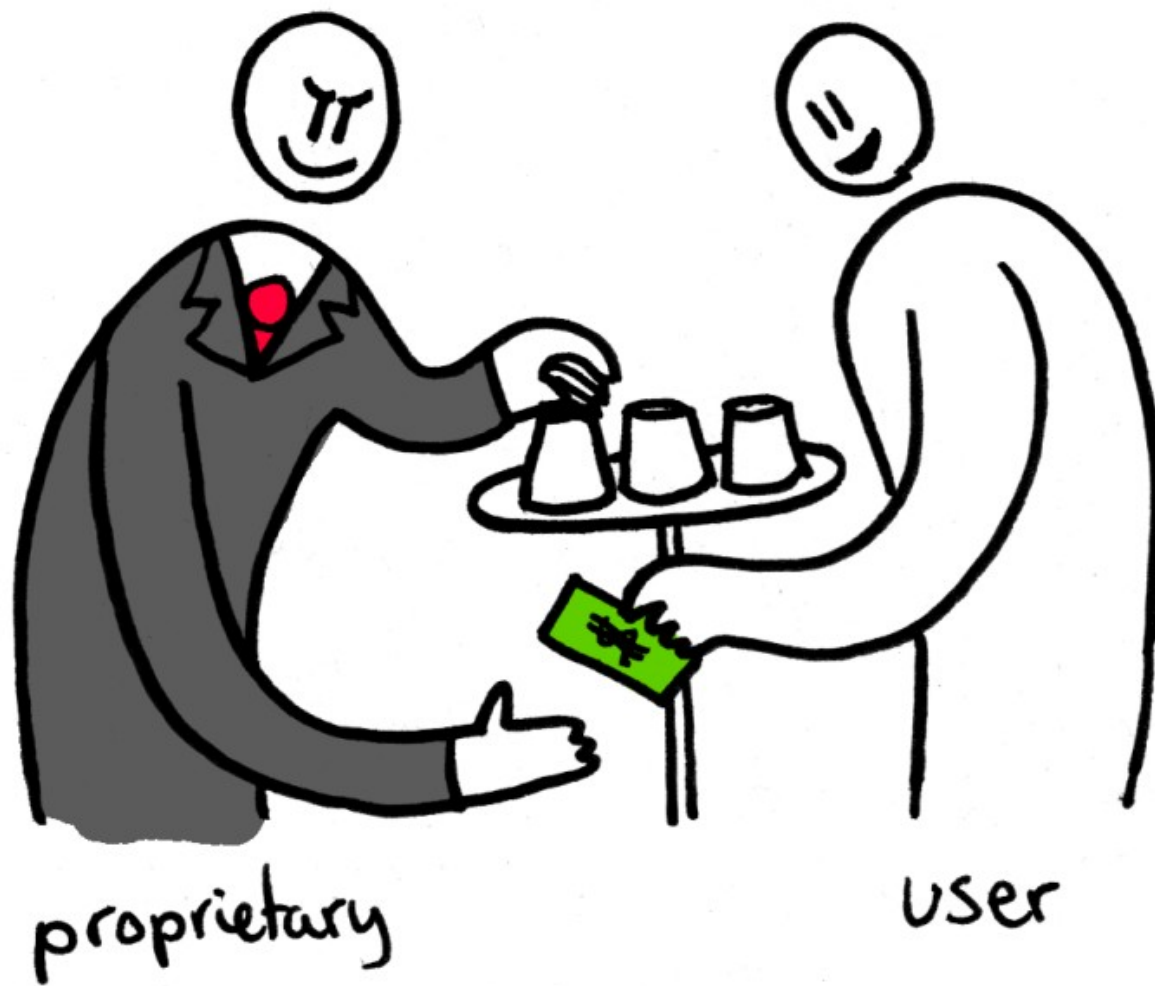
Proprietary malware

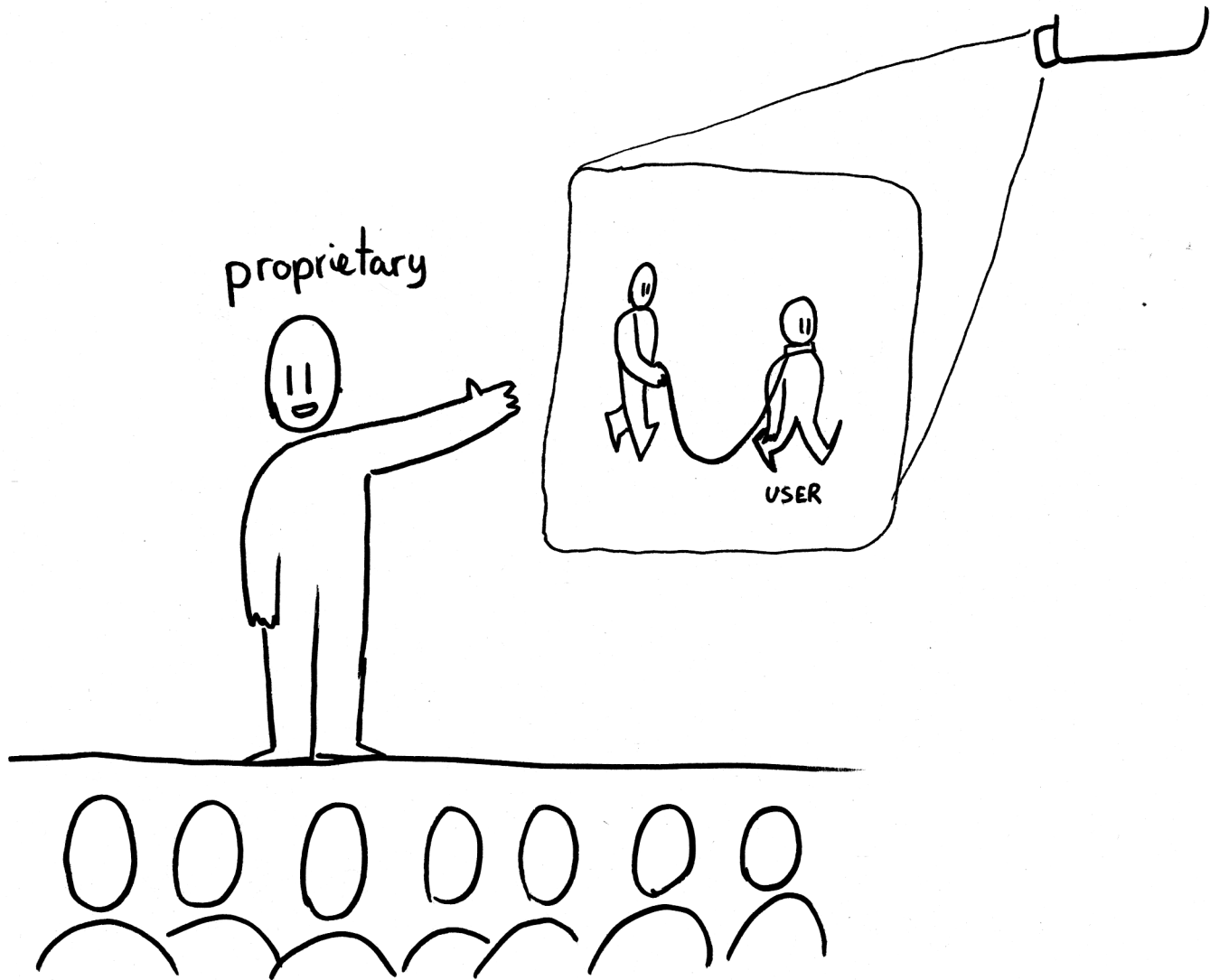


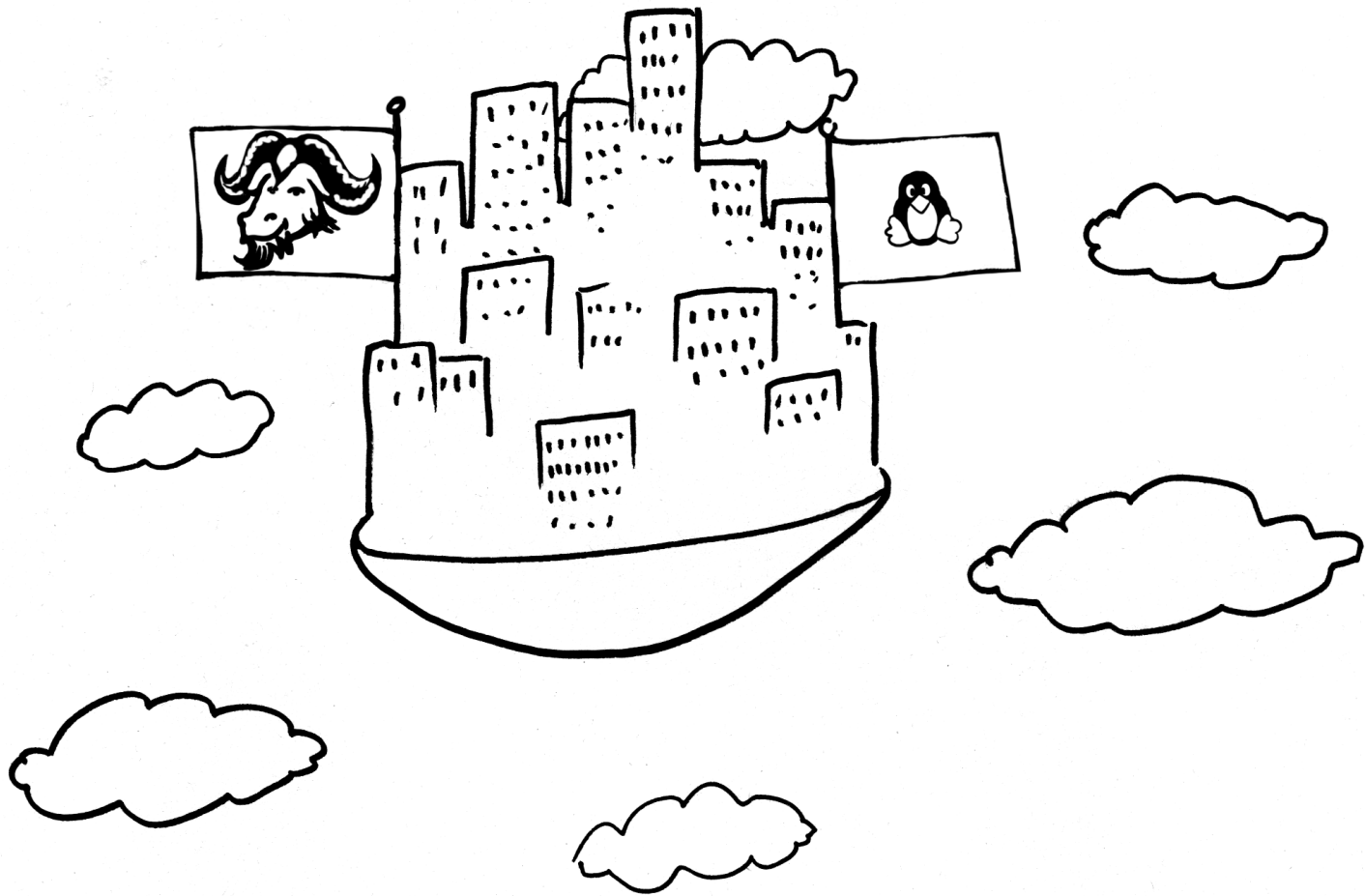


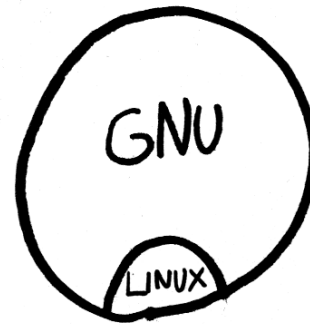
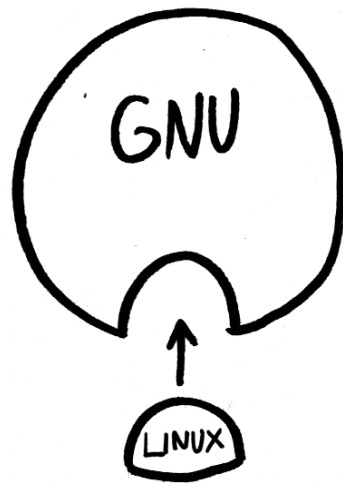
They abuse for profit



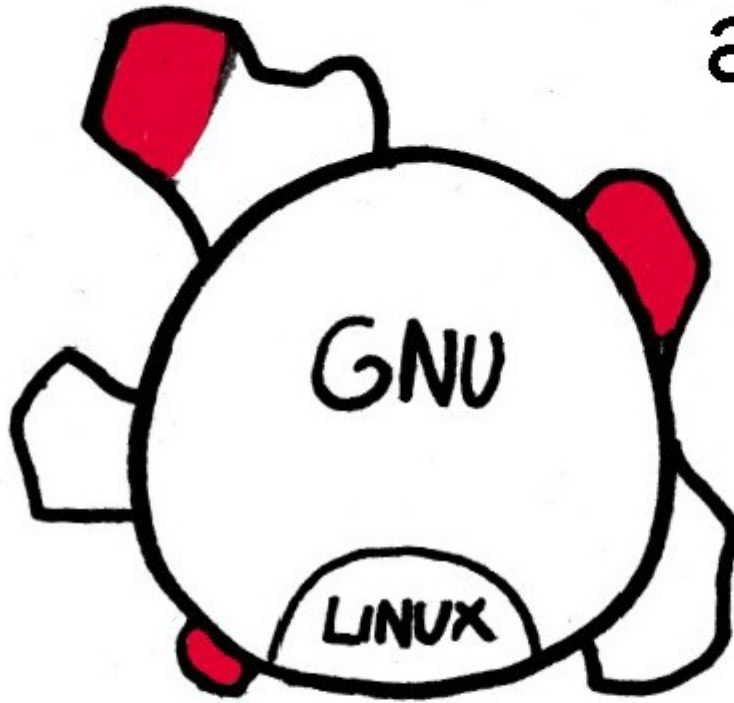








Non free applications

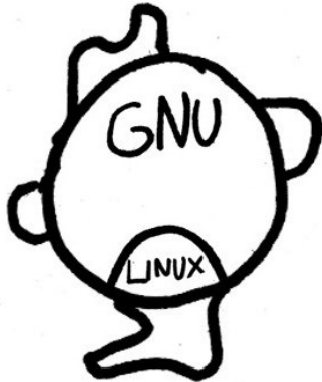


 = non free programs

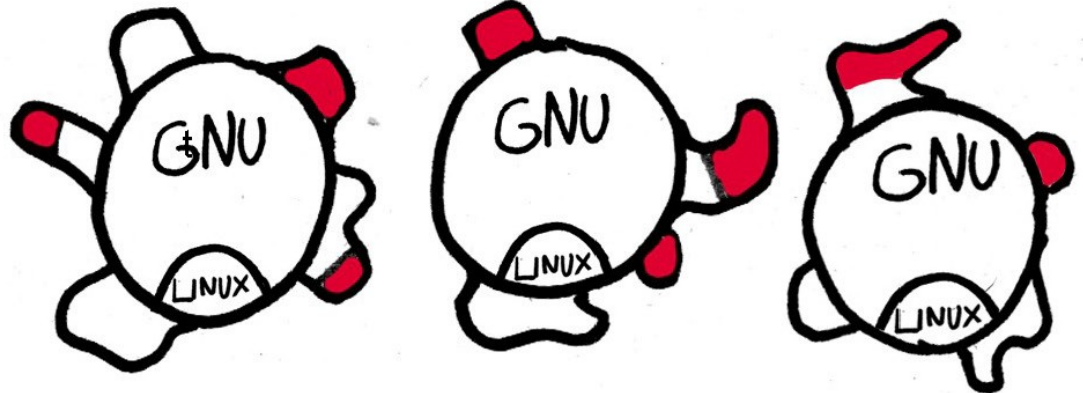


Distributions

100 % free



Mostly free

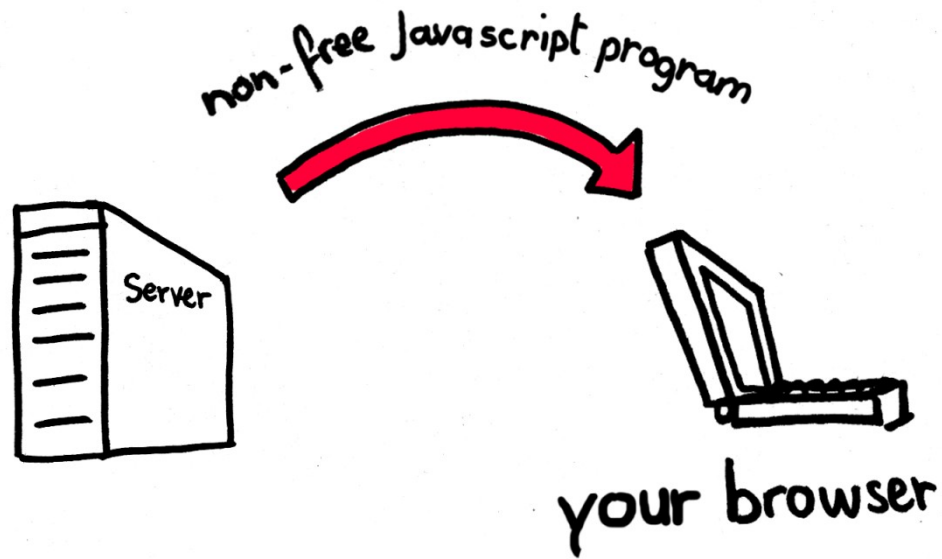


 = non free programs

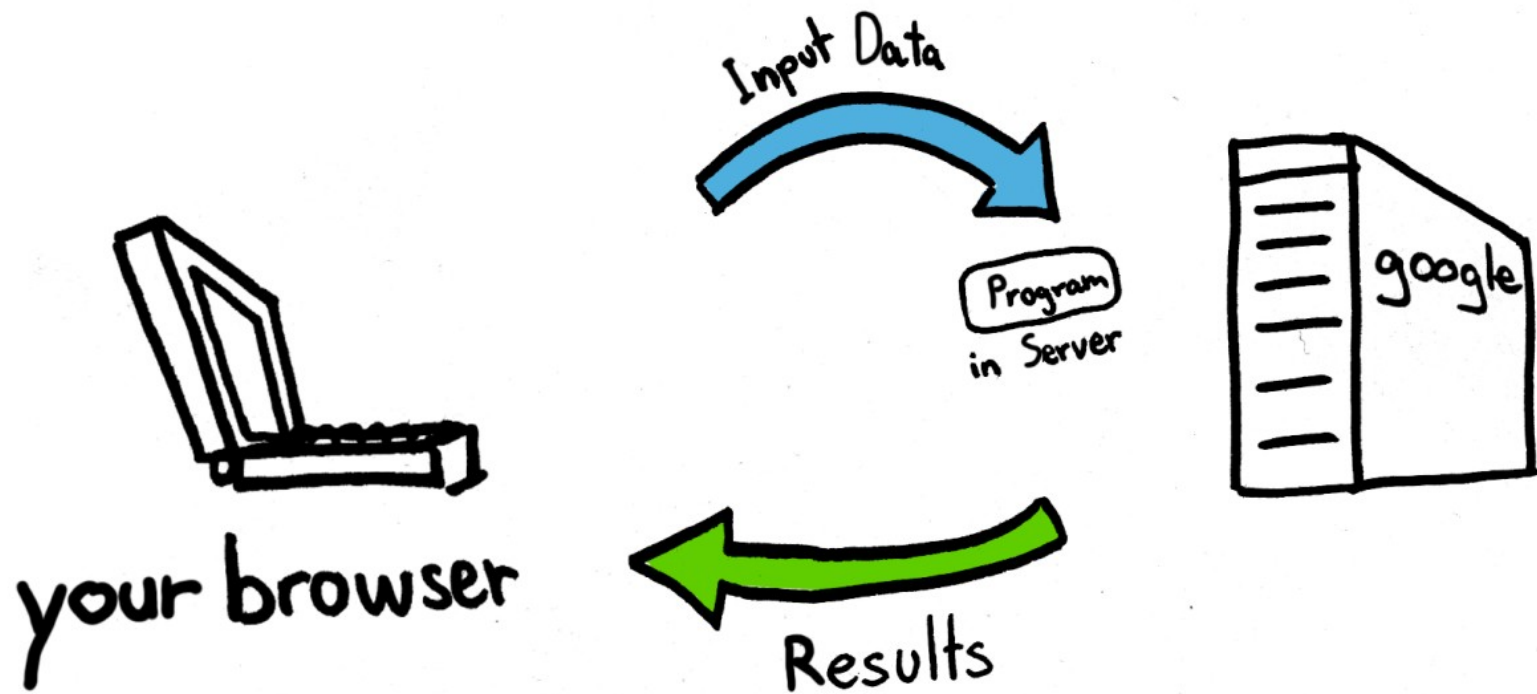


FREE SOFTWARE
FOUNDATION



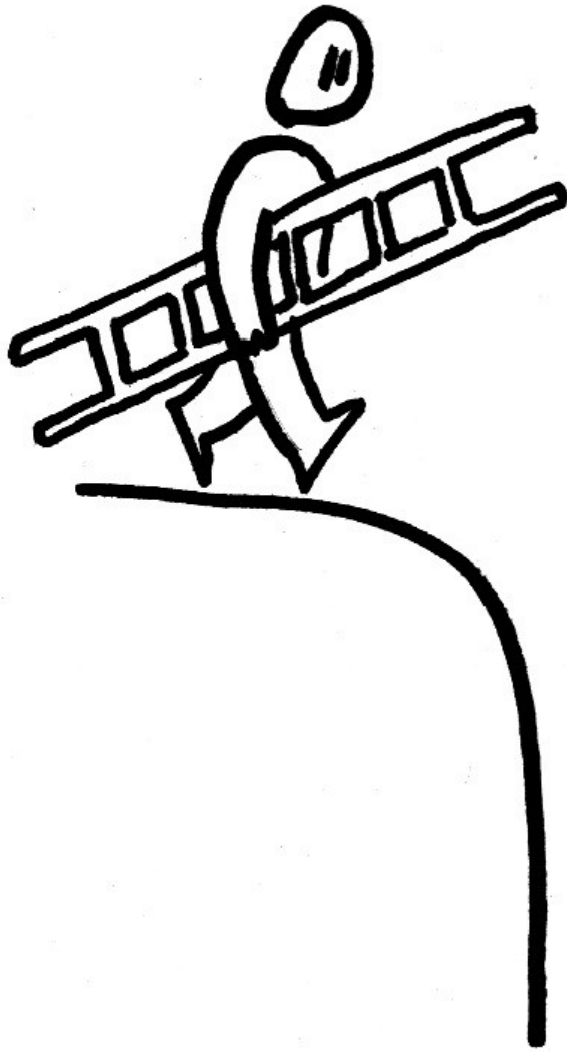


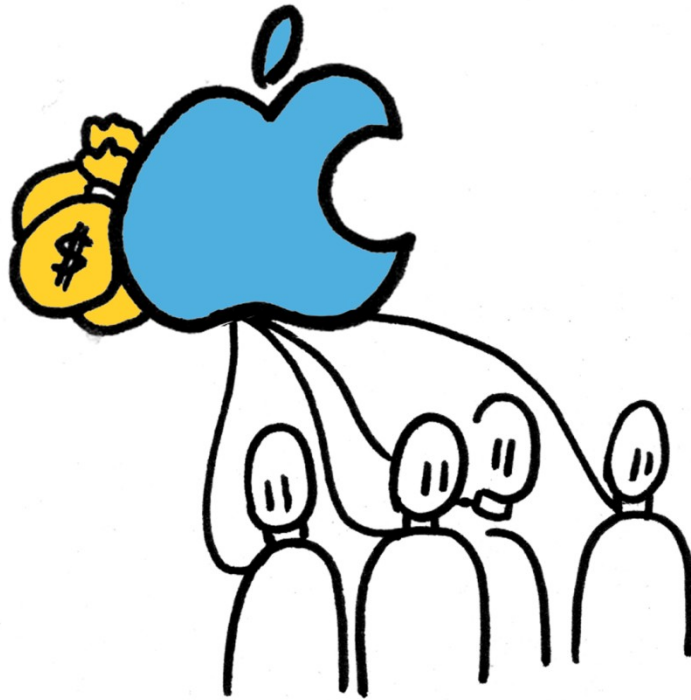
Service as a Software Substitute

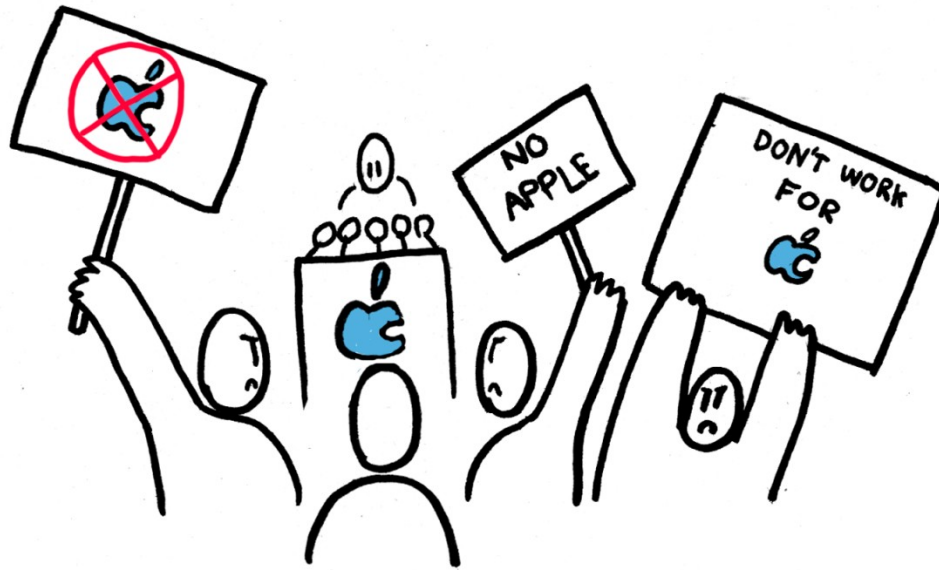




What obstacles ?



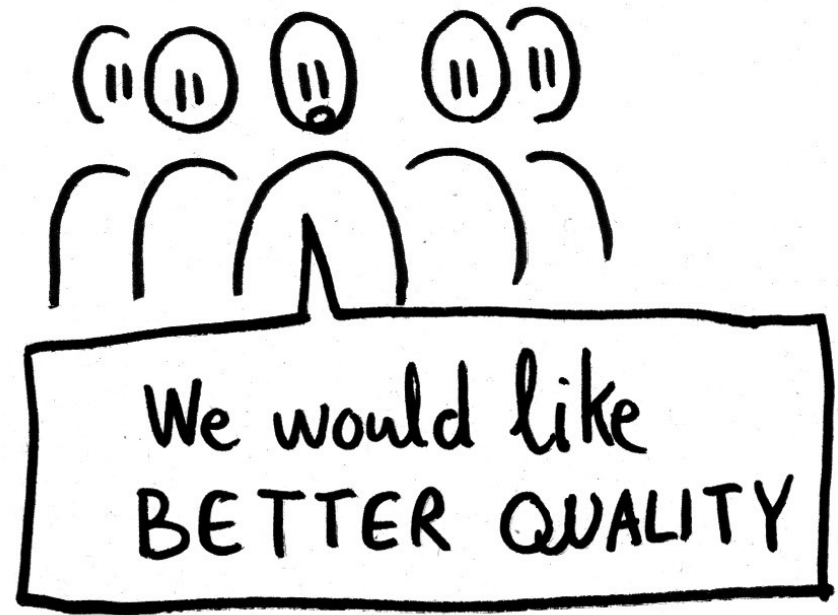


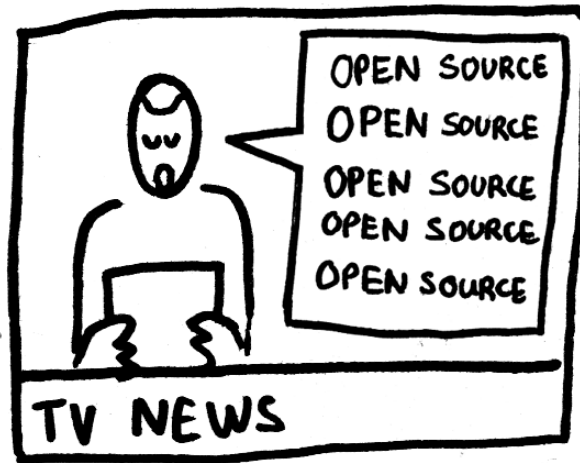


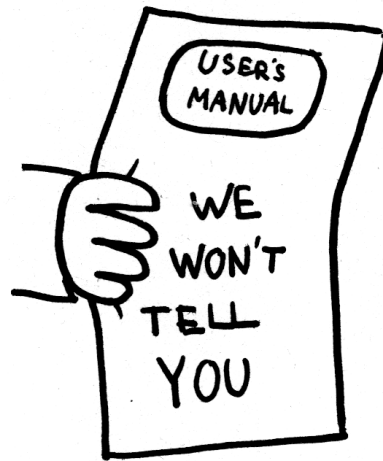
Free software activists

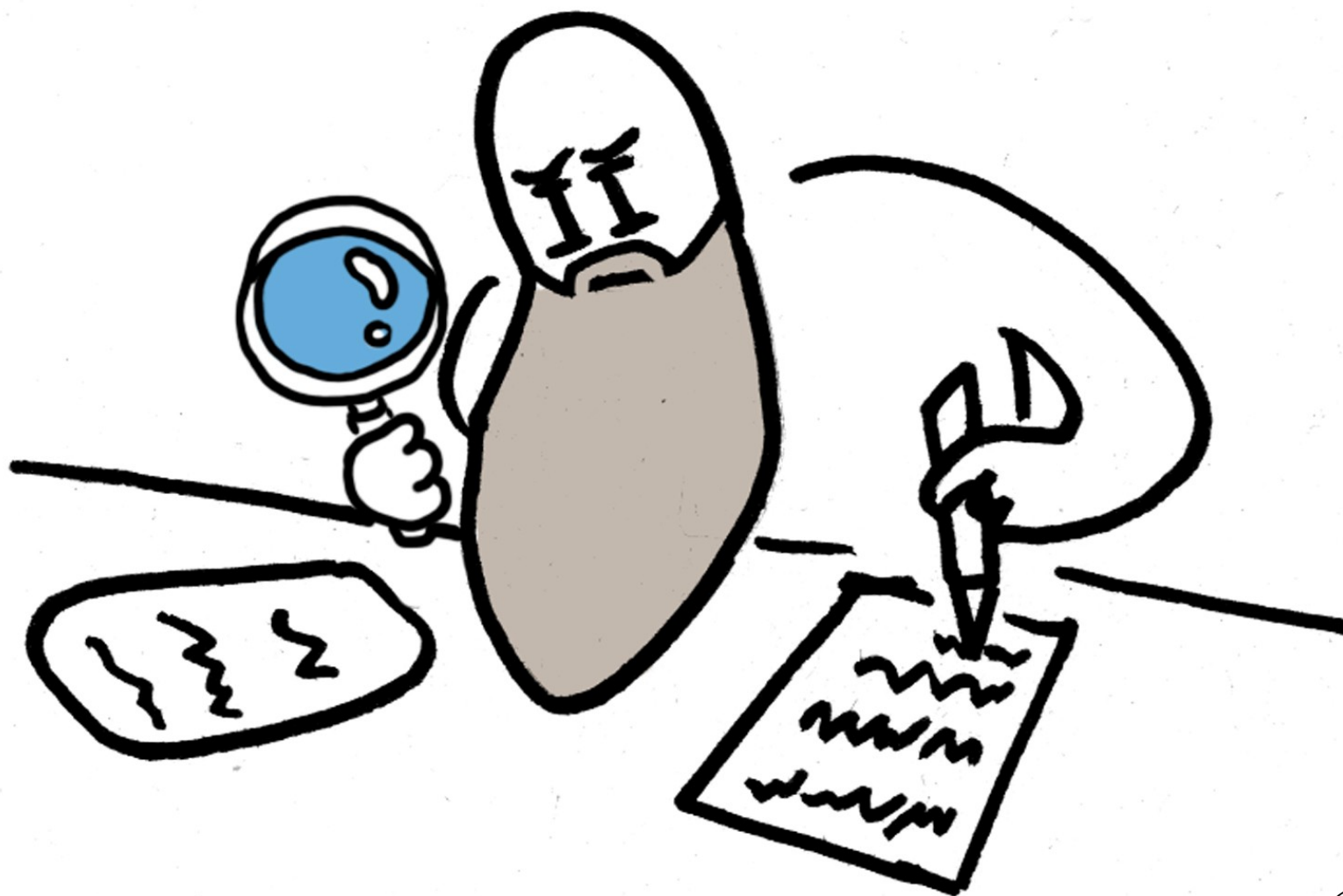


Open source boosters

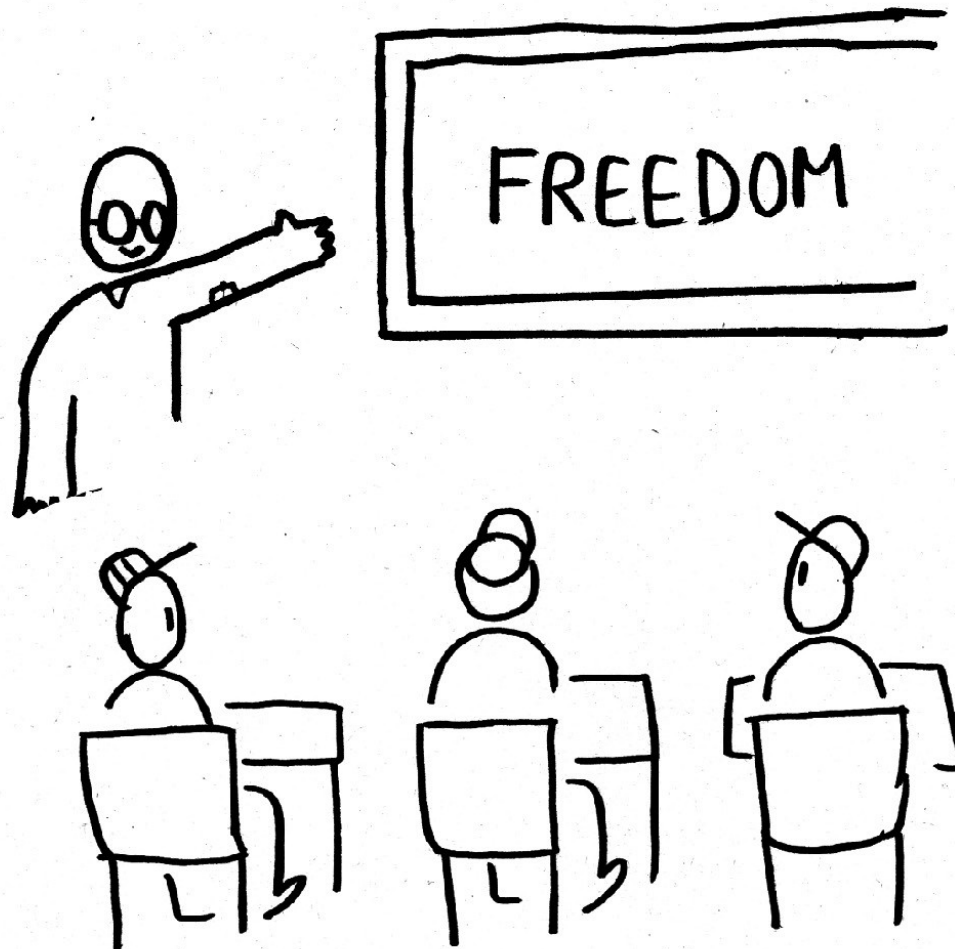


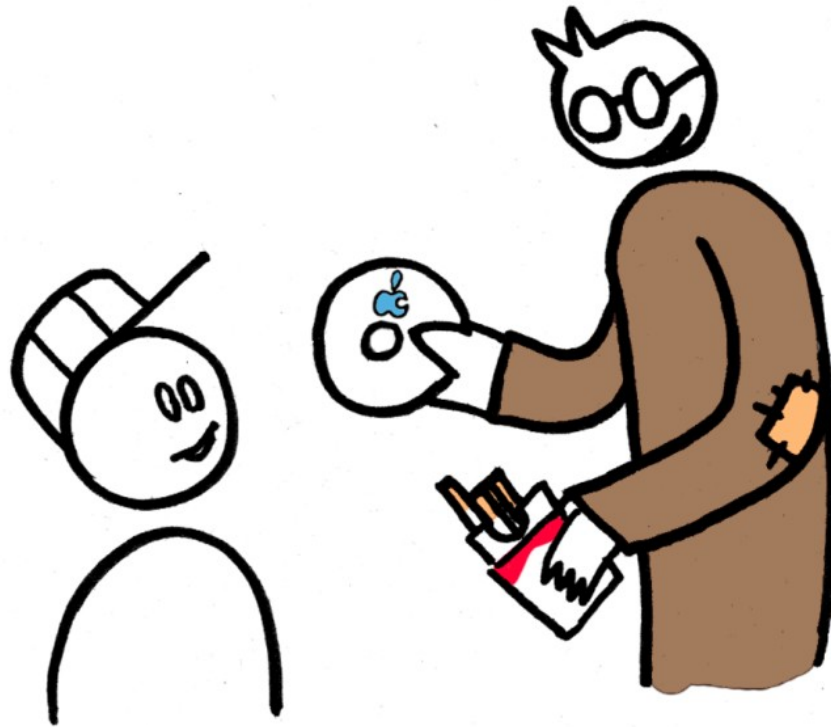






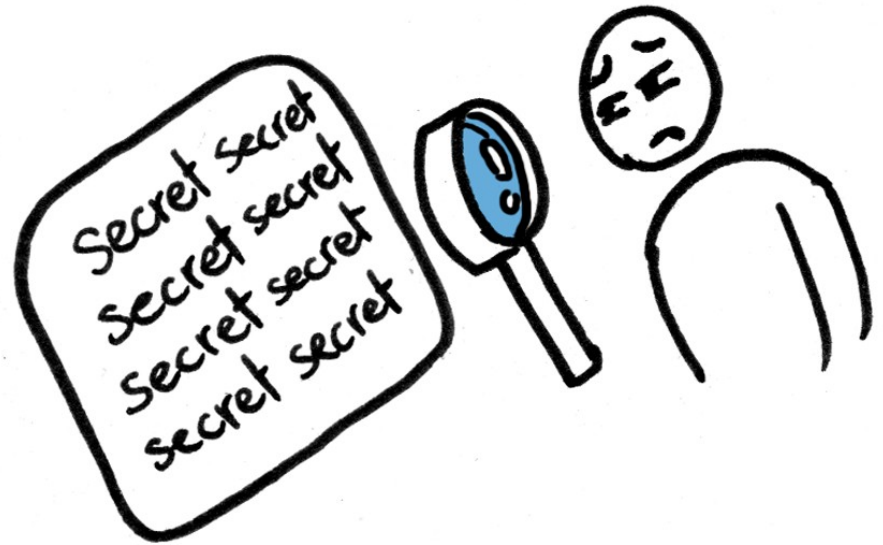
Teach free software in school



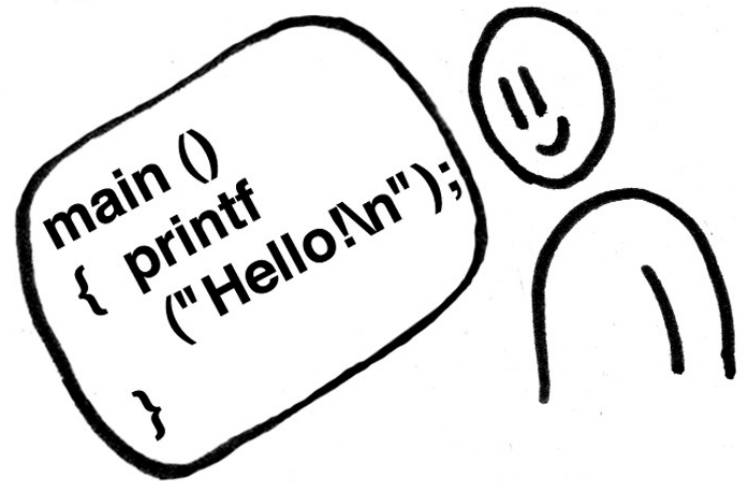


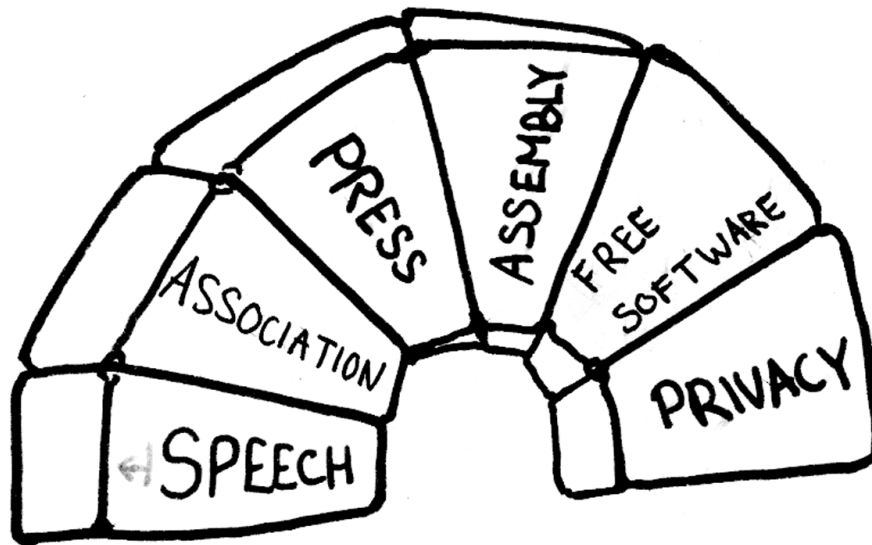


non free software:



free software:





Small sacrifices for freedom



How to help ?



Write
Free software



Organise to campaign for Free software



Persuade schools and governments



Help
other users



Say
"Free software"



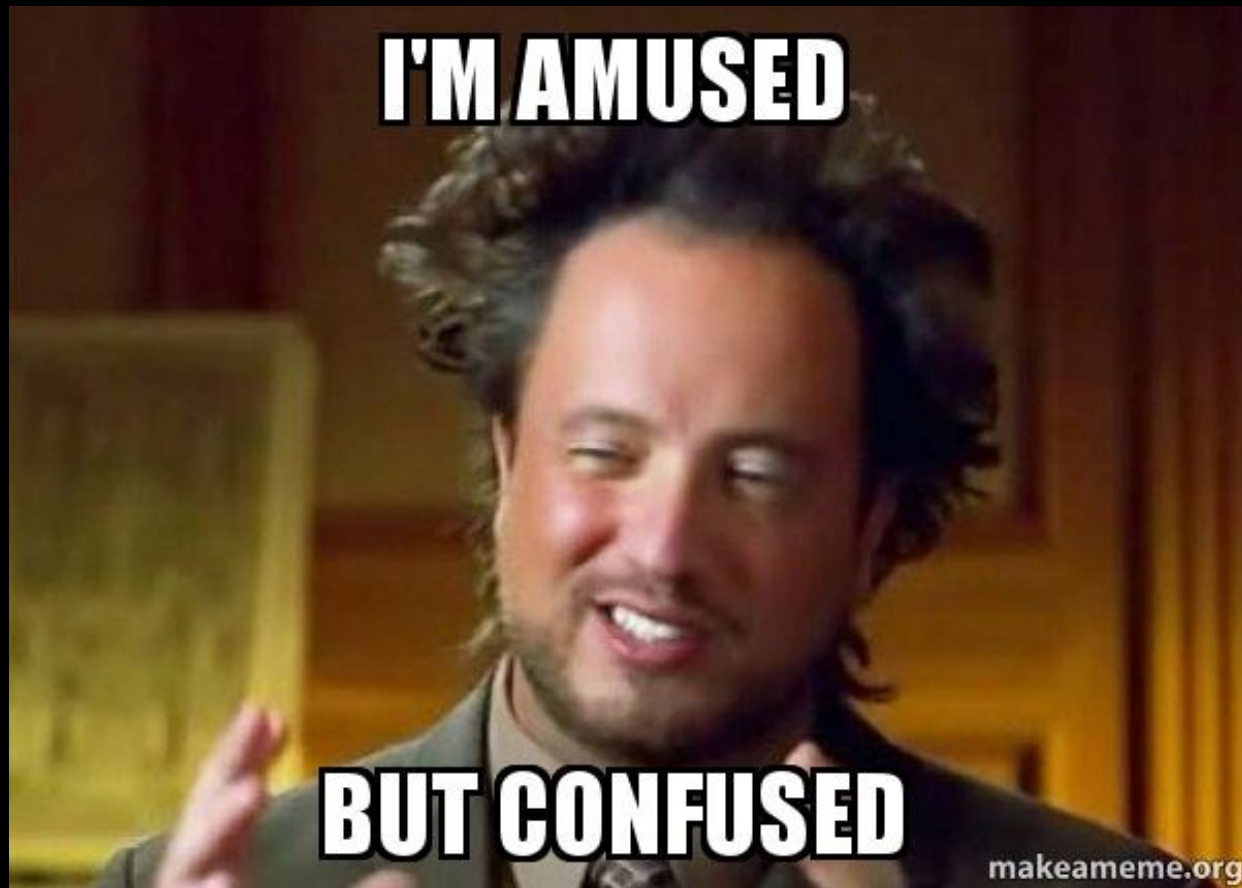
Interject

I'D JUST LIKE TO INTERJECT FOR A MOMENT.

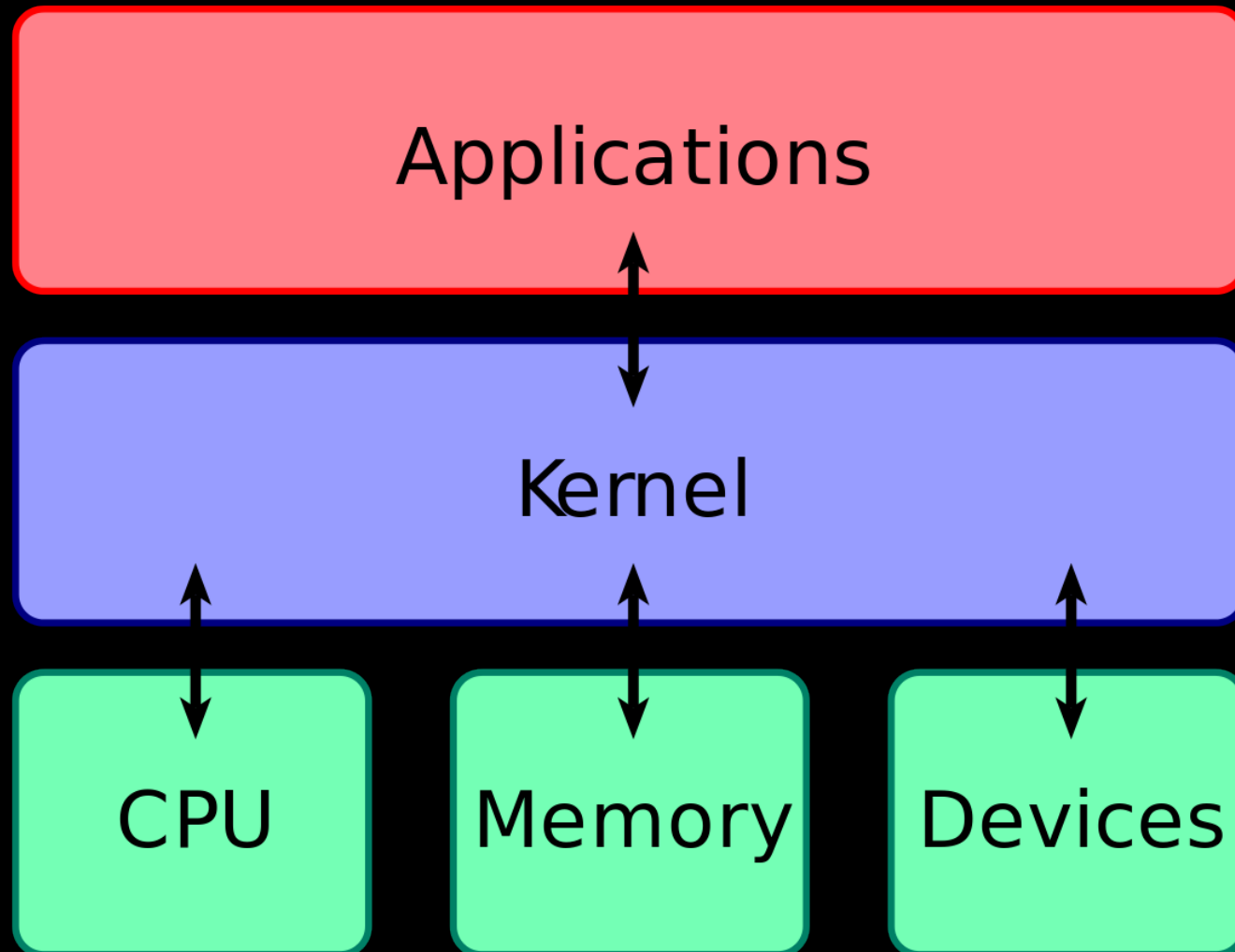
**WHAT YOU'RE REFERRING TO AS LINUX, IS IN FACT, GNU/LINUX,
OR AS I'VE RECENTLY TAKEN TO CALLING IT, GNU PLUS LINUX.**

quickmeme.com

Kernel?



Kernel



Process Scheduling



Memory Management



computer explodes

Provision of Filesystem



Creation and Termination of Processes



Access to Devices



Networking

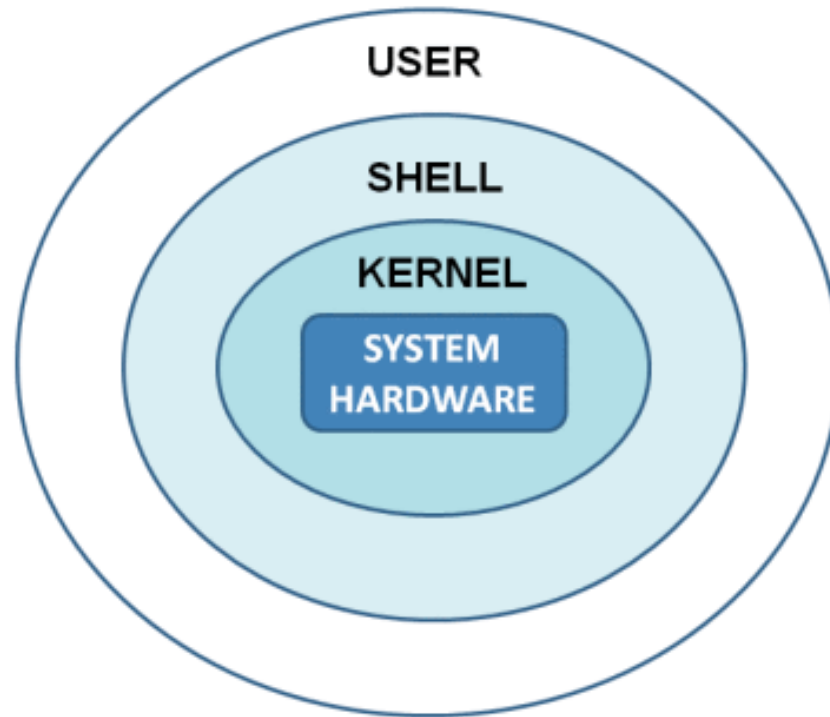
when you have 600 ping



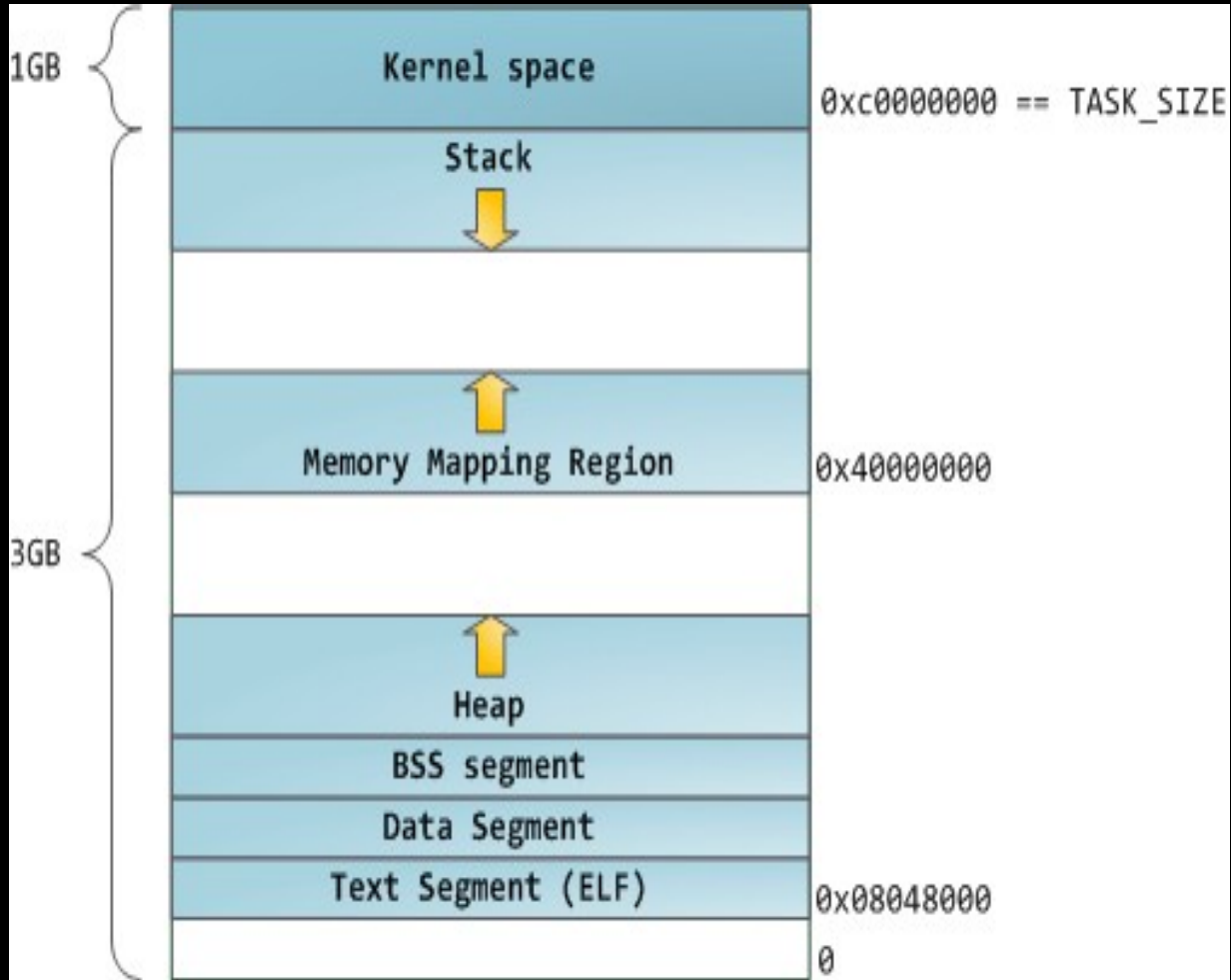
Provision of a system call application programming interface



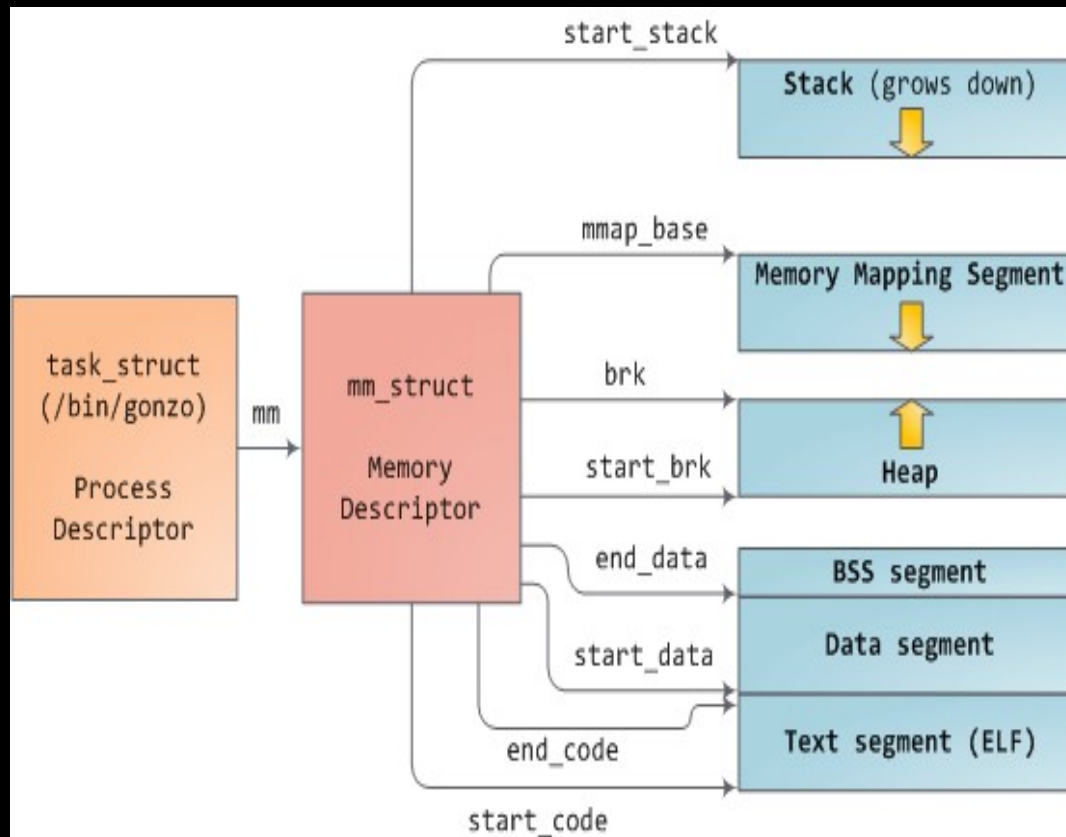
Shell



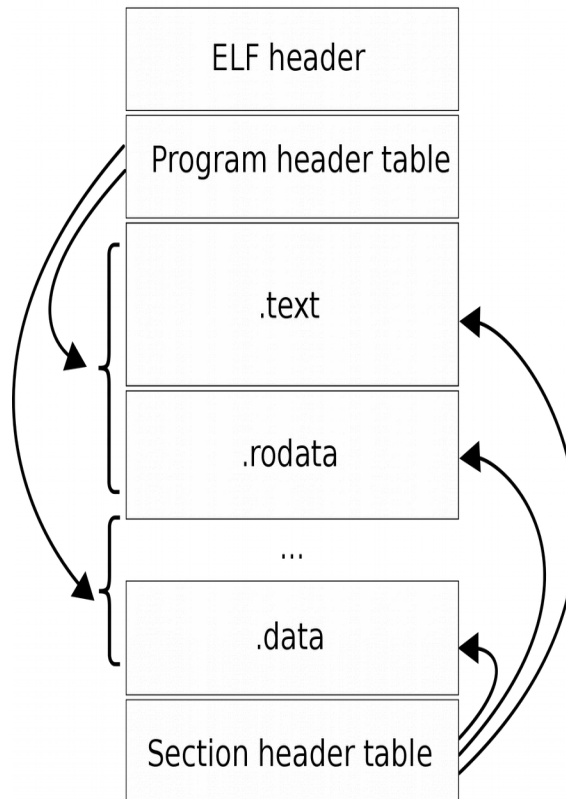
Processi



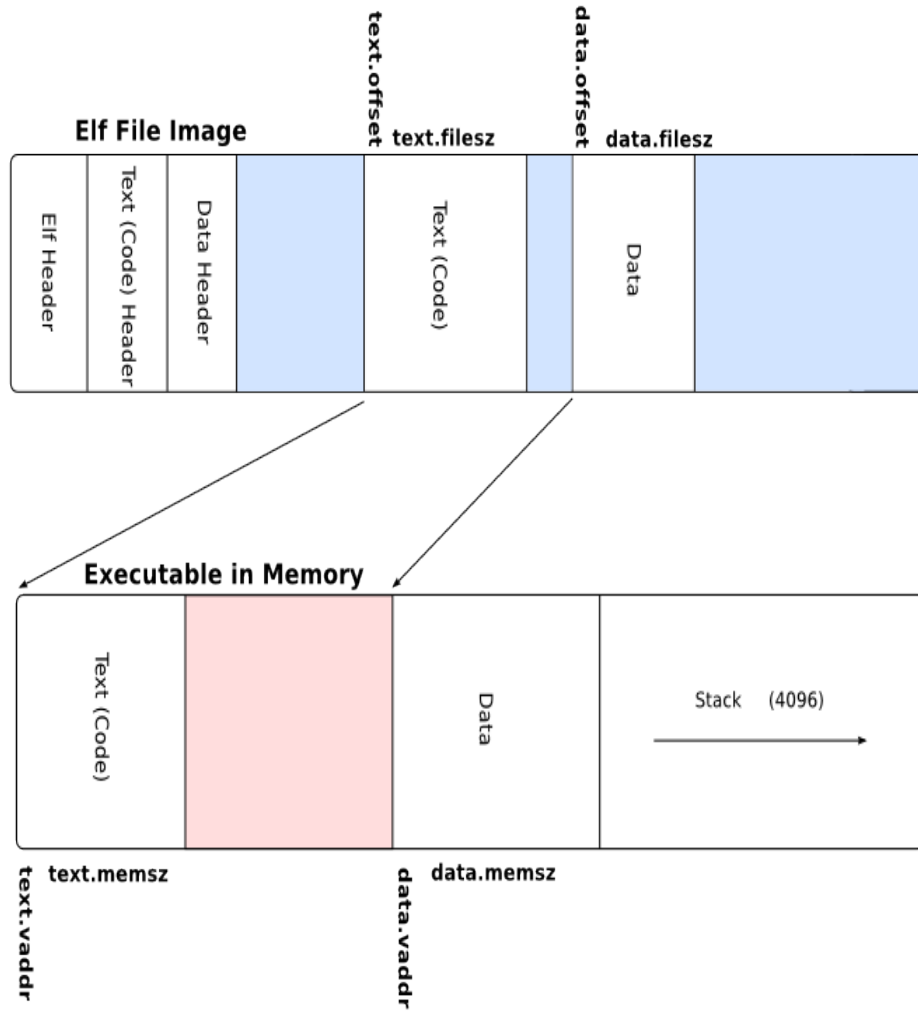
Processi



ELF



ELF



Real mode

```
C:\>dir
```

```
Volume in drive C is MS-DOS_6
```

```
Volume Serial Number is 40B4-7F23
```

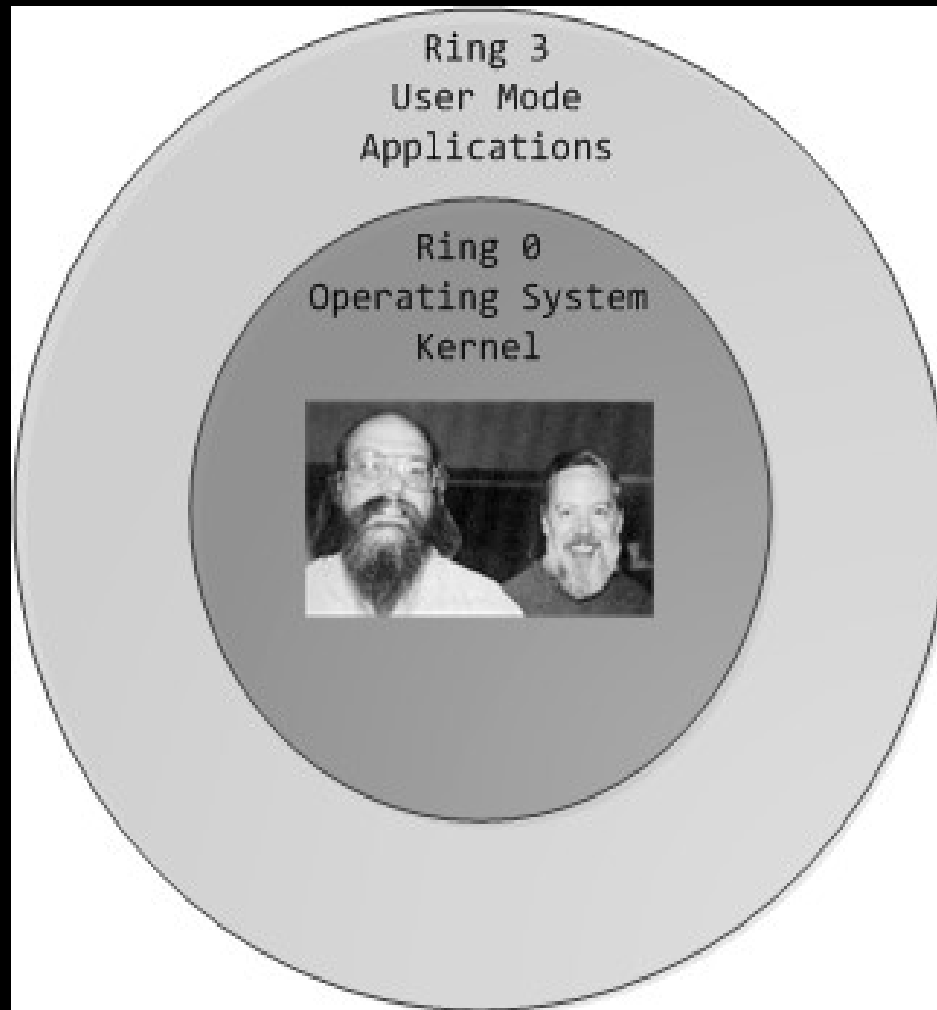
```
Directory of C:\
```

DOS	<DIR>			12.05.20	15:57
COMMAND	COM	54	645	94.05.31	6:22
WINA20	386	9	349	94.05.31	6:22
CONFIG	SYS	144		12.05.20	15:57
AUTOEXEC	BAT	188		12.05.20	15:57
5 file(s)				64 326 bytes	
				24 760 320 bytes free	

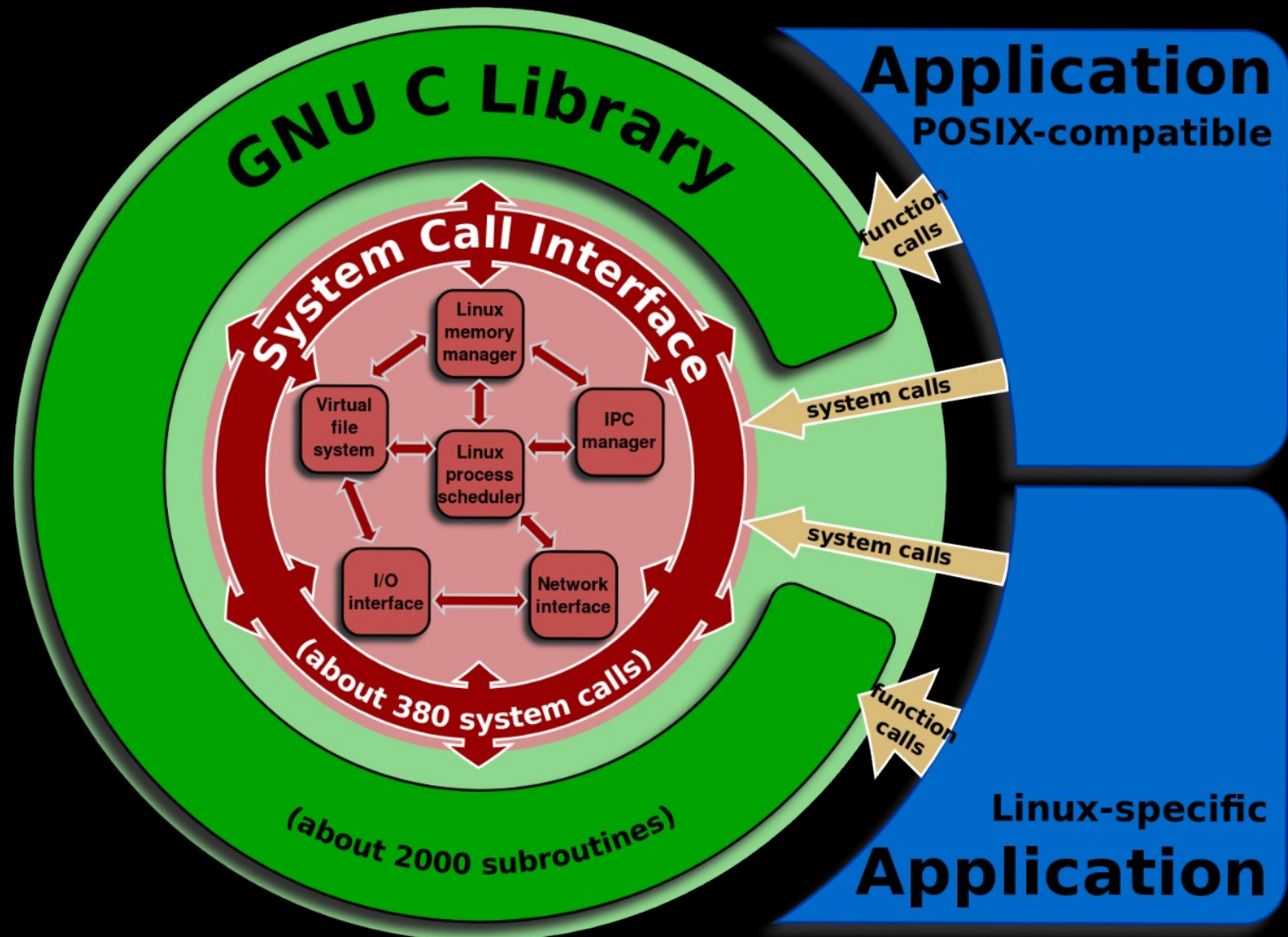
```
C:\>
```

```
C:\>  
C:\>
```

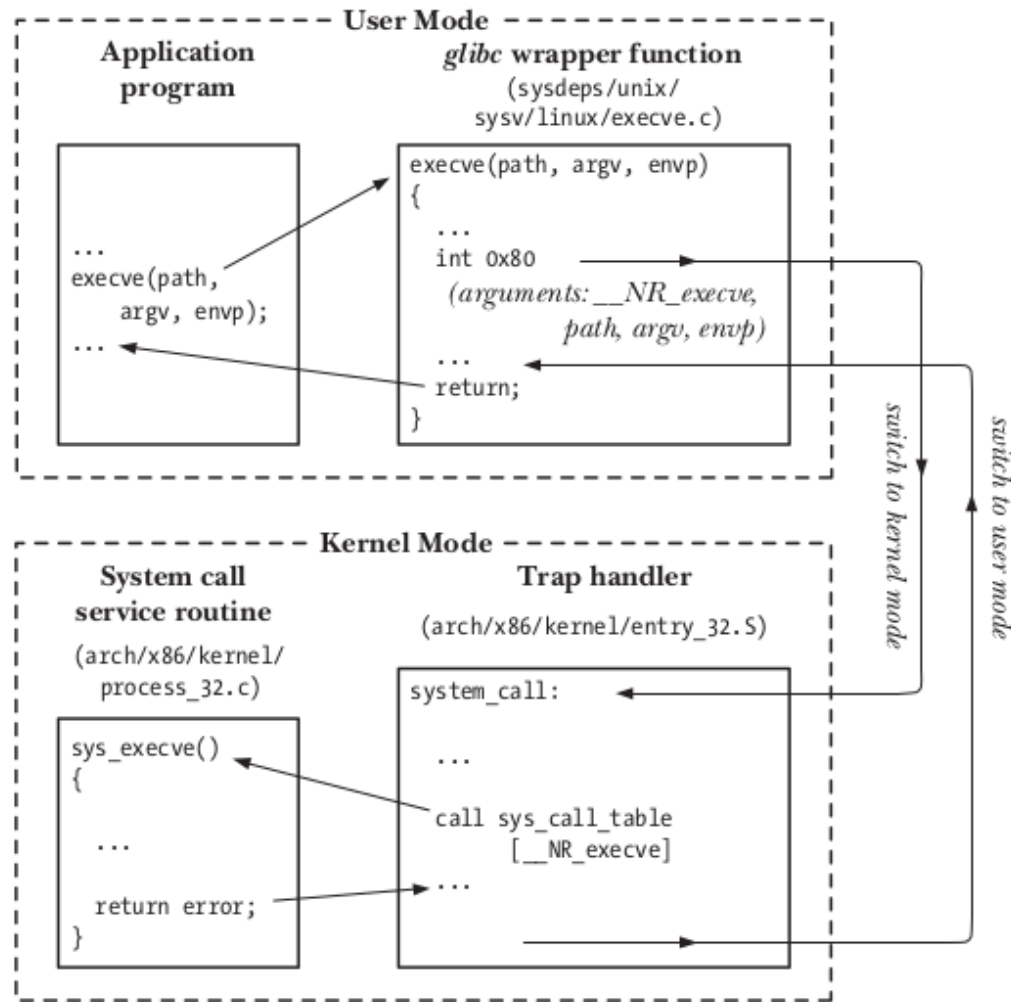
Protected mode



Syscalls



Syscalls



Library Functions

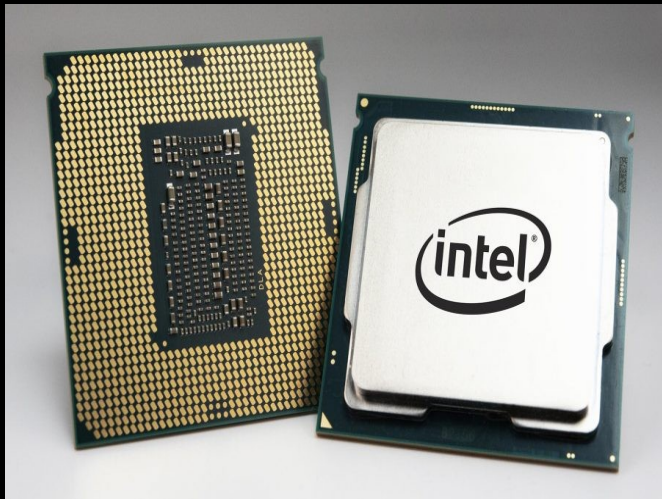
- libc
- ld
- ldd



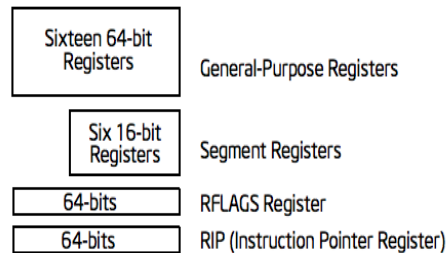
es.

- open()
- printf()
- socket()

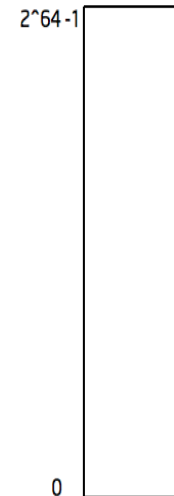
Processori



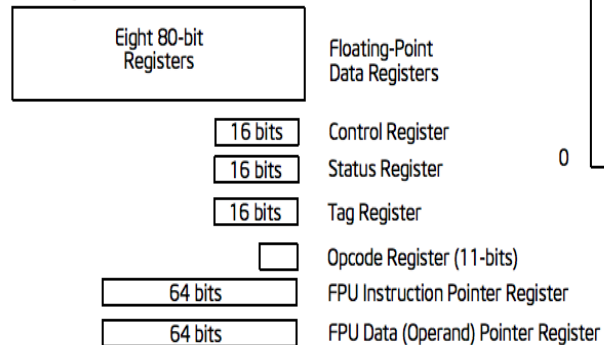
Basic Program Execution Registers



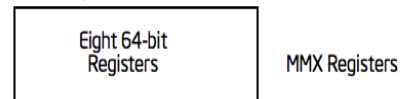
Address Space



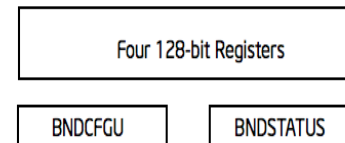
FPU Registers



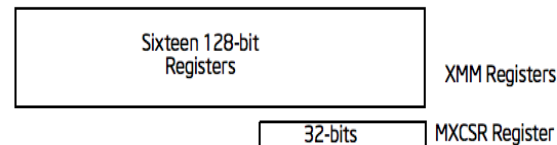
MMX Registers



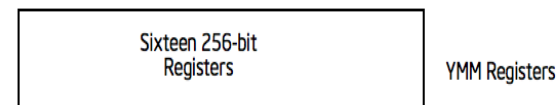
Bounds Registers



XMM Registers



YMM Registers



Assembly

```
1          SEGMENT .text
2          GLOBAL _start
3          _start:
4 00000000 B80200000      MOV RAX, 2          ; open(filename, flags)
5 00000005 48BF-         MOV RDI, filename
6 00000007 [0000000000000000]
7 0000000F 4831F6        XOR RSI, RSI          ; set flags to 0: RD_ONLY
8 00000012 0F05         SYSCALL
9 00000014 4883F8FF      CMP RAX, -1
10 00000018 7433         JE err
11 0000001A 4889C7        MOV RDI, RAX          ; file descriptor
12 0000001D 4831C0        XOR RAX, RAX          ; read(fd, buf, count)
13 00000020 48BE-         MOV RSI, buf
14 00000022 [0000000000000000]
15 0000002A BA1300000      MOV RDX, 19          ; We read at most 19 bytes
16 0000002F 0F05         SYSCALL
17 00000031 4883F8FF      CMP RAX, -1
18 00000035 7416         JE err
19 00000037 B80100000      MOV RAX, 1          ; write(fd, buf) (buf is already set from previous call)
20 0000003C BF0100000      MOV RDI, 1          ; standard output
21 00000041 0F05         SYSCALL
22 00000043 B83C00000      MOV RAX, 60         ; exit(code)
23 00000048 4831FF        XOR RDI, RDI          ; success
24 0000004B 0F05         SYSCALL
25          err:
26 0000004D B83C00000      MOV RAX, 60         ; exit(code)
27 00000052 BF0100000      MOV RDI, 1          ; 1 (fail)
28 00000057 0F05         SYSCALL
29
30          SEGMENT .bss
31          buf RESB 20
32
33          SEGMENT .rodata
34          filename DB "file",0
```

Sitografia e Bibliografia

- manybutfinite.com
- en.wikipedia.org/wiki/Executable_and_Linkable_Format
- cs.lmu.edu/~ray/notes/x86overview/
- wiki.osdev.org
- www.felixcloutier.com/x86/
- *GNU/Linux man pages*
- *The Linux Programming Interface*