COOL COMPACT OSADL ONLINE LECTURES
January 2025

OSADL cheat sheet: Technical must-knows for Open Source compliance

OSADL

Shell command line editing

Cursor to start of line	`Ctrl-A`
Cursor to end of line	`Ctrl-E`
Cut to clipboard from cursor to end of line	`Ctrl-K`
Cut to clipboard from start of line to cursor	`Ctrl-U`
Insert from clipboard to cursor	`Ctrl-Y`
Clear screen	`Ctrl-L`
Transpose current with previous character	`Ctrl-T`
Incremental reverse history search	`Ctrl-R`
Insert Ctrl-<char>	`Ctrl-V Ctrl<char>`
Uppercase from cursor to end of word	`Esc-U`
Lowercase from cursor to end of word	`Esc-L`
Cut to clipboard from cursor to end of word	`Esc-D`
Cut to clipboard from cursor to start of word	`Esc-W`

Shell commands

Change to the user’s home directory	`cd`
Change to the directory <dir>	`cd <dir>`
Change to the parent directory	`cd ..`
Change to the directory <dir> in the user’s home directory	`cd ~/dir`
Print current directory	`pwd`
Print the names of files in the current directory	`ls`
Print modes, owner, size, date and names of files in the current directory including hidden files (name starts with a dot)	`ls -al`
Same as above, order by last modified date, newest last	`ls -tral`
Print the content of the file <file>	`cat <file>`
Rename the file <old> to <new>	`mv <old> <new>`
Copy the content of file <file1> to <file2>	`cp <file1> <file2>`
Remove file <file>	`rm <file>`
Search the text snippet <pattern> in file <file>	`grep <pattern> <file>`
Search for printable text snippets in file <file>	`strings <file>`
Show content of file <file> in binary and ASCII format	`hexdump -C <file>`
Show comprehensive documentation of program <prog>	`man <prog>`
Get short help information for program <prog>	`prog --help`

Acronyms

BIOS	**B**asic **i**nput/**o**utput **s**ystem
GCC	**G**NU **c**ompiler **c**ollection
GNU	**G**nu’s **n**ot **U**nix (recursive)
GUI	**G**raphical **u**ser **i**nterface
HMI	**H**uman-**m**achine **i**nterface
KVM	**K**ernel **v**irtual **m**achine
KVM (switch)	**K**eyboard/**v**ideo/**m**ouse (switch)
RAM	**R**andom-**a**ccess **m**emory
ROM	**R**ead-**o**nly **m**emory
SMP	**S**ymmetric **m**ulti **p**rocessing (multi-core processor)
UP	**U**ni**p**rocessor (single-core processor)
UTF	**U**nicode **t**ransformation **f**ormat, e.g. UTF-8 character encoding

Git

Show overall status	`git status`

Clone upstream repository from <URL> named <name>.git, then change into the cloned directory	`git clone <URL>`
	`cd <name>`

Initialize local repo	`git init`
Show all branches	`git branch -a`
What was changed locally?	`git diff`
Show changes in relation to previous stage <ref>	`git diff <ref>`
Add changes to local repo, prepare for commit	`git add .`
Commit changes to local repo	`git commit -m "message"`
Submit local to upstream	`git push`

Was upstream updated?	`git remote update`
	`git status -uno`

Synchronize local repo with upstream	`git pull`
Show development history	`git log`
Revert all local changes (dangerous if inadvertently)	`git reset --hard`
Revert to stage <ref>	`git checkout <ref>`

Package managers: An excerpt

System or computer language	Package manager
Debian, Ubuntu Linux distributions	dpkg, apt
RedHat, Fedora Linux distributions	rpm, dnf
C / C++	Conan
Java	Maven
Rust	Cargo
Javascript	NPM
Python	Pip
PHP	Composer

Derivation

Interface between...	and...	Derivation
user program	Linux kernel	no
network client	network server	no
shared memory of program #1	memory access from program #2	no
forking program	forked program	no
program source code	modified version of the source code	**YES**
program calling a function	a function in another source code	**YES**
program calling a function	a function in another executable	**YES**
program calling a function	a function in a plugin	**YES**

Build commands

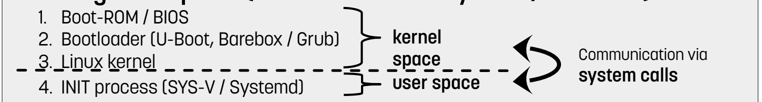
Classic	`./configure`
	`make`
	`sudo make install`

Cmake	`mkdir _build`
	`cd _build`
	`cmake ..`
	`make; sudo make install`

Meson	`mkdir _build`
	`meson setup _build`
	`meson -C _build compile`
	`sudo meson -C _build install`

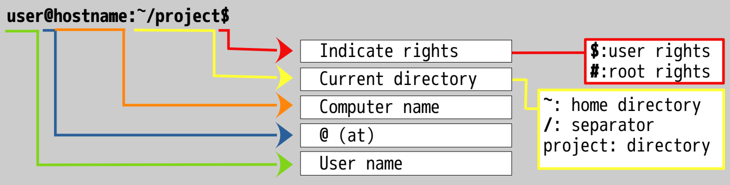
Provided by: Open Source Automation Development Lab (OSADL) eG, Heidelberg, Germany – www.osadl.org, info@osadl.org
OSADL supports using Open Source software in industry with respect to technical and legal aspects. OSADL is a registered cooperative and accepts members from everywhere in the world.

Booting a computer (Linux embedded system / Linux PC)



SecureBoot verifies the integrity of each component before starting it.

Terminal prompt (default configuration)



Command line

```
user@hostname:~/project$ command [options] [arguments]
```

command: The program or command to be executed (e.g. ls, echo, cd).
options: Additional modifiers that change the behavior of the command
arguments: File name or data to which the command is applied

Directory tree

```
user
├── data
│   ├── case1
│   │   ├── file1
│   │   └── file2
│   └── case2
├── info.txt
├── project
│   └── sheet.tab
└── work
```

Parent directory of **data**, info.txt, **project** and **work**

The *directory separator* is **/**, e.g. the path of this file is: /user/data/case1/file2

The top-level directory of a computer system is called *root directory.*

Bits and Bytes: Binary and hexadecimal notation

bit: a single storage cell that can be 0 or 1 (**binary** notation)
Byte: 8 bits, can represent numbers from 0 to 255

Hexadecimal notation:
0 -> 0 1 -> 1 2 -> 2 3 -> 3 4 -> 4 5 -> 5 6 -> 6 7 -> 7
8 -> 8 9 -> 9 10 -> A 11 -> B 12 -> C 13 -> D 14 -> E 15 -> F

The number 181 in **binary**, **decimal** and **hexadecimal** notation:

$$1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad = \mathbf{10110101}$$

$$\underbrace{1*2^7 + 0*2^6 + 1*2^5 + 1*2^4}_{1*2^3 + 0*2^2 + 1*2^1 + 1*2^0} + \underbrace{0*2^3 + 1*2^2 + 0*2^1 + 1*2^0}_{0*2^3 + 1*2^2 + 0*2^1 + 1*2^0} = \mathbf{181}$$

$$\underbrace{1*2^3 + 0*2^2 + 1*2^1 + 1*2^0}_{11 \to B} \quad \underbrace{0*2^3 + 1*2^2 + 0*2^1 + 1*2^0}_{5 \to 5} \quad = \mathbf{B5}$$

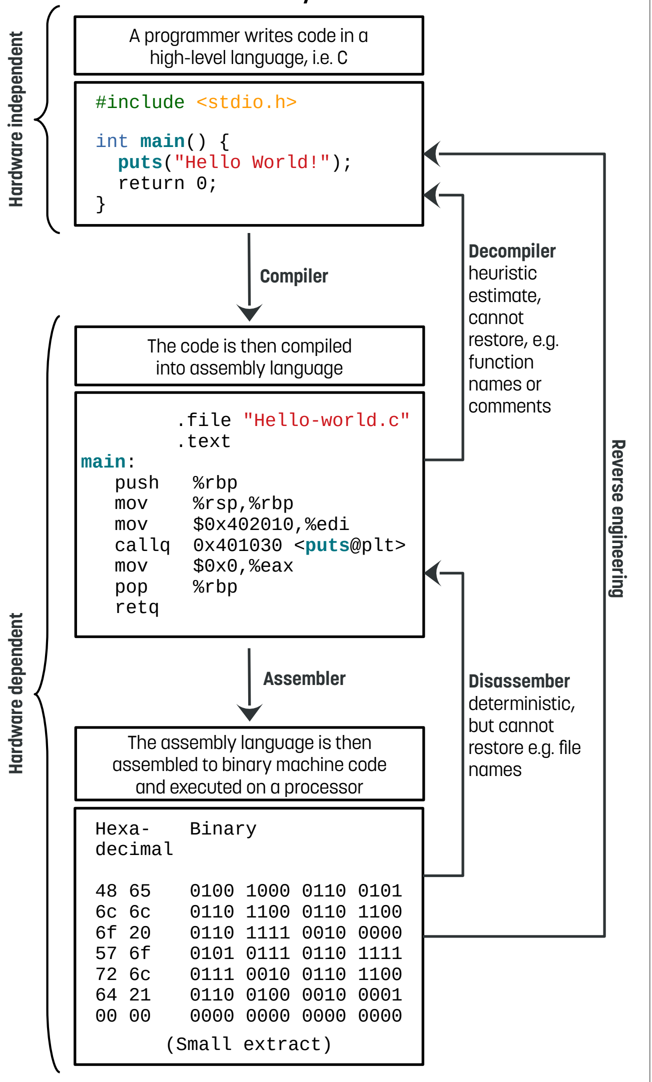
String encoding: ASCII

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
Non printable control symbols								**BS**		**LF**			**CR**		
10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
Non printable control symbols															
20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F
	"	#	$	%	&	'	(	)	*	+	,	-	.	/	
30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F
P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F
`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F
p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	

High-level computer languages (examples)

	Interpreter languages	Compiler languages
General purpose	PHP, Javascript, Python	C/C++, C#, Rust, Java
Problem oriented	APL, R	Fortran, Cobol

From source code to binary executable (and back)



A programmer writes code in a high-level language, i.e. C

```
#include <stdio.h>

int main() {
  puts("Hello World!");
  return 0;
}
```

The code is then compiled into assembly language

```
        .file "Hello-world.c"
        .text
main:
    push    %rbp
    mov     %rsp,%rbp
    mov     $0x402010,%edi
    callq   0x401030 <puts@plt>
    mov     $0x0,%eax
    pop     %rbp
    retq
```

The assembly language is then assembled to binary machine code and executed on a processor

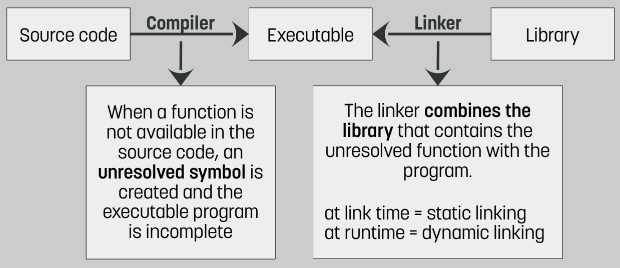
```
Hexa-        Binary
decimal

48 65        0100 1000 0110 0101
6c 6c        0110 1100 0110 1100
6f 20        0110 1111 0010 0000
57 6f        0101 0111 0110 1111
72 6c        0111 0010 0110 1100
64 21        0110 0100 0010 0001
00 00        0000 0000 0000 0000
        (Small extract)
```

Decompiler heuristic estimate, cannot restore, e.g. function names or comments

Disassembler deterministic, but cannot restore e.g. file names

Link dependencies

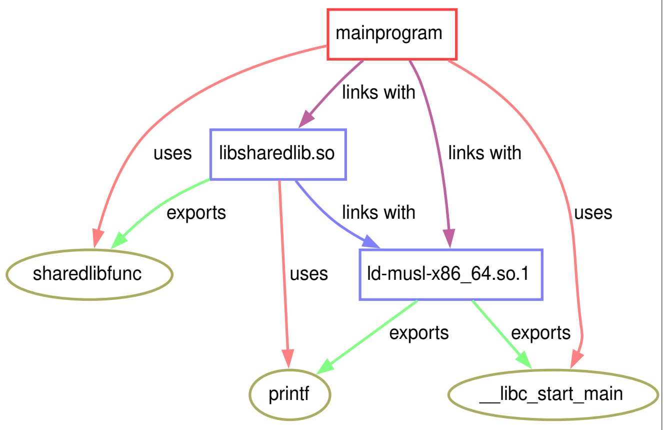


When a function is not available in the source code, an **unresolved symbol** is created and the executable program is incomplete

The linker **combines the library** that contains the unresolved function with the program.

at link time = static linking
at runtime = dynamic linking

Callgraph of a program with libraries and their symbols



V1 - January, 2025