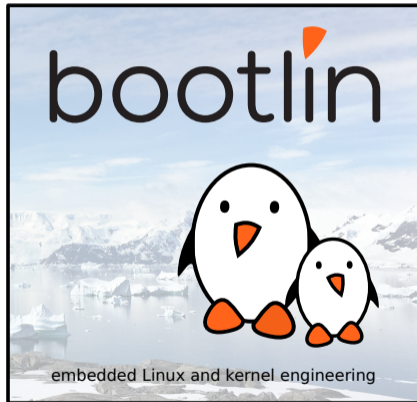




## Embedded Linux nuggets found in Buildroot package Eldorado

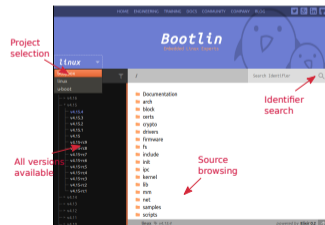
Michael Opdenacker  
*michael.opdenacker@bootlin.com*

© Copyright 2004-2021, Bootlin.  
Creative Commons BY-SA 3.0 license.  
Corrections, suggestions, contributions and translations are welcome!





- ▶ Founder and Embedded Linux engineer at Bootlin:
  - ▶ Embedded Linux **expertise**
  - ▶ **Development**, consulting and training
  - ▶ Focusing **only on Free and Open Source Software**
- ▶ Free Software contributor:
  - ▶ Current maintainer of the Elixir Cross Referencer, making it easier to study the sources of big C projects like the Linux kernel. See <https://elixir.bootlin.com>
  - ▶ Current documentation maintainer for the Yocto Project
  - ▶ Co-author of Bootlin's freely available embedded Linux and kernel training materials (<https://bootlin.com/docs/>)
  - ▶ Former maintainer of **GNU Typist**





# Buildroot package Eldorado

Buildroot 2021.08 supports about 2850 packages, estimated by the number of `.mk` files.

Section	Toplevel menu entries
Audio and video applications	52
Compressors and decompressors	14
Debugging, profiling and benchmark	68
Development tools	37
Filesystem and flash utilities	42
Fonts, cursors, icons, sounds and themes	17
Games	22
Graphic libraries and applications (graphic/text)	93
Hardware handling	185
Interpreter languages and scripting	652
Libraries (all types)	677
Mail	8
Miscellaneous	27
Networking applications	268
Package managers	3
Real-time	2
Security	15
Shell and utilities	37
System tools	96
Text editors and viewers	9



## Approach

- ▶ Tried going through all the packages in `make menuconfig`, but it made me overlook many interesting packages.
- ▶ Instead, checked the more recent new packages in git commits (going only 2 years back in time)
- ▶ Warning: the selection of components is completely arbitrary. It also depends on the topics I'm familiar with, and I could be missing other cool nuggets.
- ▶ I listed the components I found in random order, so that you don't get bored by categories you're not interested in.

## Tips

- ▶ Only Free and Open Source software was selected in this presentation.
- ▶ For each package, the license is easy to find in `package/<package>/<package>.mk`



## Contribute to the public pad

I'm going to show my own selection of "cool" packages in my opinion

- ▶ During the presentation (even if you are watching it afterwards), let us know about useful tools that you discovered and would deserve to be better known.
- ▶ Do this by contributing to this Etherpad:  
<https://annuel2.framapad.org/p/buildroot-nuggets-9pmi?lang=en>
- ▶ We'll be able to check the suggestions on the pad at the end of the talk.



A simple TTY terminal I/O application

<https://tio.github.io/>

- ▶ Added on 2019-12-22
- ▶ Even simpler than Picocom
- ▶ Doesn't die when your device is disconnected. Just waits and resumes nicely.
- ▶ Easy command line (no line settings):  
`tio /dev/ttyUSB0`
- ▶ Nice timestamping option

```
lundmar@wopr: ~  
lundmar@wopr:~$ tio /dev/serial/by-id/usb-FTDI_FT232R_USB_UART-if00-port0  
[tio 17:25:27] tio v1.23  
[tio 17:25:27] Press ctrl-t q to quit  
[tio 17:25:27] Connected  
$  
$ # We are now connected to a TTY terminal via a usb->serial cable  
$  
$ uname -snri  
Linux hummingboard 3.14.60 Freescale i.MX6 Quad/DualLite (Device Tree) GNU/Linux  
$  
$ # Lets try unplug and plug usb cable and see what happens..  
$  
[tio 17:25:44] Disconnected  
[tio 17:25:46] Connected  
$  
$ # Tio automatically reconnected  
$  
$ # Tio supports various useful key commands  
$  
$ # Press ctrl-t ? to list available key commands  
$  
$
```

<https://tio.github.io/images/tio-demo.gif>



Tool for flashing block devices efficiently and reliably  
From Artem Bityutskiy

<https://github.com/intel/bmap-tools>

- ▶ Added on 2021-06-21
- ▶ Originally created for the *Tizen IVI* project.
- ▶ Can be much faster than `dd` and `cp`, taking advantage of unmapped areas (*holes*).
- ▶ Can directly flash an image from a remote server without downloading it first.
- ▶ Implements integrity checking. Corruption will be noticed immediately.

Tool to flash image files to block devices using the block map. `bmaptool` is a generic tool for creating the block map (bmap) for a file, and copying files using the block map. The idea is that large file containing unused blocks, like raw system image files, can be copied or flashed a lot faster with `bmaptool` than with traditional tools like "`dd`" or "`cp`".

casync — A tool for distributing file system images

<https://github.com/systemd/casync>

- ▶ Added on 2021-01-10
- ▶ By Lennart Poettering (Systemd, PulseAudio...)
- ▶ Targeting the delivery of OS images. Could be used for backup purposes too.
- ▶ Combines the idea of the rsync algorithm with the idea of git-style content-addressable file systems.
- ▶ The main goals are to minimize network traffic and disk space on both clients and servers.

#### Operations on blob index files

```
# casync digest --store=/var/lib/backup.castr fedora25.ca1bx
# casync mdev --store=/var/lib/backup.castr fedora25.ca1bx
# casync verify --store=/var/lib/backup.castr fedora25.ca1bx /home/lennart/Fedora25.raw (NOT IMPLEMENTED)
```

#### Operations involving ssh remoting

```
# casync make foobar:/srv/backup/lennart.ca1dx /home/lennart
# casync extract foobar:/srv/backup/lennart.ca1dx /home/lennart2
# casync l1st foobar:/srv/backup/lennart.ca1dx
# casync digest foobar:/srv/backup/lennart.ca1dx
# casync ntree foobar:/srv/backup/lennart.ca1dx
# casync mount foobar:/srv/backup/lennart.ca1dx /home/lennart
```

#### Operations involving the web

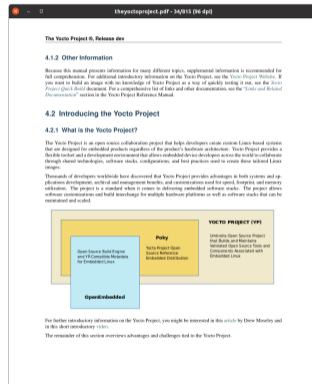
```
# casync extract http://www.foo-bar.com/lennart.ca1dx /home/lennart
# casync l1st http://www.foo-bar.com/lennart.ca1dx
# casync digest http://www.foo-bar.com/lennart.ca1dx
# casync ntree http://www.foo-bar.com/lennart.ca1dx
# casync extract --seed=/home/lennart http://www.foo-bar.com/lennart.ca1dx /home/lennart2
# casync mount --seed=/home/lennart http://www.foo-bar.com/lennart.ca1dx /home/lennart2
```



## Lightweight PDF, XPS, and E-book (EPUB) viewer

<https://www.mupdf.com/>

- ▶ Added on 2020-02-07
- ▶ Very fast and producing high quality output
- ▶ Should be suitable for portable viewers
- ▶ Some issues with some EPUB documents though





A new binary file editor - <http://www.jemarch.net/poke>

GNU poke is an interactive, extensible editor for binary data. Not limited to editing basic entities such as bits and bytes, it provides a full-fledged procedural, interactive programming language designed to describe data structures and to operate on them.

- ▶ Added on 2021-05-17
- ▶ Time to ditch your hexadecimal editor!
- ▶ Ready made *pickles* are available to process many types of binary formats.

```
(poke) dump
76543210 0011 2233 4455 6677 8899 aabb ccdd eeff 0123456789ABCDEF
00000000: 7f45 4c46 0201 0100 0000 0000 0000 0000 .ELF.....
00000010: 0100 3e00 0100 0000 0000 0000 0000 0000 ..>.....
00000020: 0000 0000 0000 0000 0802 0000 0000 0000 .....
00000030: 0000 0000 4000 0000 0000 4000 0b00 0a00 ....@.....@....
00000040: 5548 89e5 b800 0000 005d c300 4743 433a UH.....]..GCC:
00000050: 2028 4465 6269 616e 2036 2e33 2e30 2d31 (Debian 6.3.0-1
00000060: 382b 6465 6239 7531 2920 362e 332e 3020 8+deb9u1) 6.3.0
00000070: 3230 3137 3035 3136 0000 0000 0000 0000 20170516.....
(poke) load elf
(poke) var ehdr = Elf64_Ehdr @ 0#B
(poke) ehdr.e_ident
struct {
  ei_mag=[0x7fUB,0x45UB,0x4cUB,0x46UB],
  ei_class=0x2UB,
  ei_data=0x1UB,
  ei_version=0x1UB,
  ei_osabi=0x0UB,
  ei_abiversion=0x0UB,
  ei_pad=[0x0UB,0x0UB,0x0UB,0x0UB,0x0UB,...],
  ei_nident=0x0UB#B
}
```



Easy to embed functional and imperative programming language

<https://janet-lang.org/>

- ▶ Added on 2021-02-13
- ▶ Makes a good system scripting language, or a language to embed in other programs, like Lua and Guile. It has more built-in functionality and a richer core language than Lua, but is smaller than GNU Guile or Python.
- ▶ Easy to embedded: just need `janet.c` and `janet.h`

```
(defn sum3
  "Solve the 3SUM problem in O(n^2) time."
  [s]
  (def tab @[])
  (def solutions @())
  (def len (length s))
  (for k 0 len
    (put tab (s k) k))
  (for i 0 len
    (for j 0 len
      (def k (get tab (- 0 (s i) (s j))))
      (when (and k (not= k 1) (not= k j) (not= 1 j))
        (put solutions [1 true j true k true] true))))
  (map keys (keys solutions)))

(let [arr @[2 4 1 3 8 7 -3 -1 12 -5 -8]]
  (printf "3sum of %j: " arr)
  (printf "%j" (sum3 arr)))
```



## A dynamic tracer for Linux

<https://wkz.github.io/ply/>

- ▶ Added on 2021-01-06
- ▶ Very simple to deploy: no need for kernel modules, no compiler, just need a libc and a kernel with BPF support.
- ▶ Very low overhead, can be used even for "the hottest code paths"
- ▶ C like syntax to describe what to trace and output. Scripts can be written and modified in seconds.

### read(2) return distribution

```
#!/usr/bin/env ply
kretprobe:Sys_read
{
    @[ "retsize" ] = quantize(retval);
}
```

This example shows a very simple script that instruments the return of the `read(2)` syscall and records the distribution of the return argument.

User-defined aggregations in ply always start with `@`. When writing small scripts, only using one aggregation, it is common and convenient to simply call it `@`.

```
~$ sudo ./read-dist.ply
ply: active
^Cply: deactivating
```

```
@:
{ retsize ):
    < 0      861 | ██████████
    [ 0, 1]  865 | ██████████
    [ 2, 3]  372 | ████████
    [ 4, 7]   1 |
    [ 8, 15] 1690 | ██████████
    [ 16, 31] 122 | ████████
    [ 32, 63]  22 |
    [ 64, 127] 25 |
    [ 128, 255] 1 |
    ...
    [ 512, 1k]  23 |
    [ 1k, 2k]   5 |
    ...
    [ 4k, 8k]   2 |
    ...
    [ 16k, 32k] 11 |
```



# Neofetch

Bash script to display system information

<https://github.com/dylananaraps/neofetch>

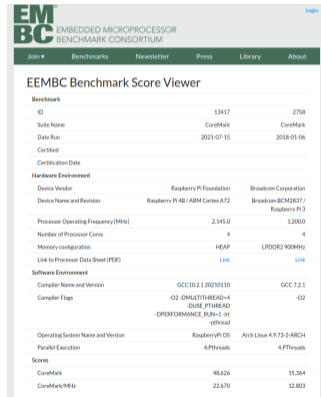
- ▶ Added on 2021-01-15
- ▶ This displays system information, including hardware, in a visually pleasing way
- ▶ Meant to be used in screenshots of your system
- ▶ You can also add its output to your system logs
- ▶ 2019: already supporting 150 different operating systems

```
      .-/+oosssso+/-.      mike@mike-laptop
      `:+ssssssssssssss+:`
      -+ssssssssssssssyyssss+
      .ossssssssssssssdMMMMy sssso.
      /ssssssssshdmmNNmyNMMMMh sssss/
      +ssssssshnydMMMMMMNdddyssssss+
      /ssssssshNMMMyhhyyyhdNMMMMh sssss/
      .sssssssdMMMNhssssssshNMMMd sssss.
      +sssxhhhyNMMNy ssssssssssyNMMMy sssss+
      ossyNMMMNyMMh sssssssssshmmh sssssso
      ossyNMMMNyMMh sssssssssshmmh sssssso
      +sssxhhhyNMMNy ssssssssssyNMMMy sssss+
      .sssssssdMMMNhssssssshNMMMd sssss.
      /ssssssshNMMMyhhyyyhdNMMMMh sssss/
      +sssssssdnydMMMMMMNdddyssssss+
      /ssssssssshdmmNNmyNMMMMh sssss/
      .ossssssssssssssdMMMMy sssso.
      -+ssssssssssssssyyssss+
      `:+ssssssssssssss+:`
      .-/+oosssso+/-.      OS: Ubuntu 20.04.3 LTS x86_64
      -----
      Host: Latitude E7450
      Kernel: 5.4.0-81-generic
      Uptime: 2 days, 21 hours, 53 mins
      Packages: 3761 (dpkg), 19 (snap)
      Shell: bash 5.0.17
      Resolution: 1920x1080, 1920x1080
      DE: GNOME
      WM: Mutter
      WM Theme: Adwaita
      Theme: Yaru [GTK2/3]
      Icons: Yaru [GTK2/3]
      Terminal: terminator
      CPU: Intel i7-5600U (4) @ 3.200GHz
      GPU: Intel HD Graphics 5500
      Memory: 4988MiB / 7850MiB
      [Red][Green][Yellow][Blue][Purple][Cyan][Grey]
```

## Microcontroller and CPU benchmarking tool

<https://www.eembc.org/coremark/>

- ▶ Added on 2020-11-12
- ▶ Allows to compare CPUs and MCUs in terms read/write, integer, control operations and pipeline performance.
- ▶ Small enough to fit in the CPU cache
- ▶ May be useful if you hesitate between several CPUs for your design.



**EEMBC** EMBEDDED MICROPROCESSOR BENCHMARK CONSORTIUM

Join ▾ Benchmarks Newsletter Press Library About

### EEMBC Benchmark Score Viewer

<b>Benchmark</b>		
ID	13417	2758
Suite Name	CoreMark	CoreMark
Date Run	2021-07-15	2018-01-06
Certified		
Certification Date		
<b>Hardware Environment</b>		
Device Vendor	Raspberry Pi Foundation	Broadcom Corporation
Device Name and Revision	Raspberry Pi 4B / ARM Cortex A72	Broadcom BCM2837 / Raspberry Pi 3
Processor Operating Frequency (MHz)	2,145.0	1,200.0
Number of Processor Cores	4	4
Memory configuration	HEAP	LPDDR2 900MHz
Link to Processor Data Sheet (PDF)	<a href="#">Link</a>	<a href="#">Link</a>
<b>Software Environment</b>		
Compiler Name and Version	GCC 10.2.1 20210110	GCC 7.2.1
Compiler Flags	-O2 -DMULTITHREAD=4 -DUSE_PTHREAD -DPERFORMANCE_RUN=1 -t -pthread	-O2
Operating System Name and Version	RaspberryPi OS	Arch Linux 4.9.73-2-ARCH
Parallel Execution	4Pthreads	4Pthreads
<b>Scores</b>		
CoreMark	48,626	15,364
CoreMark/MHz	22,670	12,803

Small and embeddable Javascript engine  
<https://bellard.org/quickjs/>

- ▶ Added on 2020-11-06
- ▶ Yet another project from Fabrice Bellard (QEMU, FFMPEG, TCC...)
- ▶ Small and easily embeddable: just a few C files, no external dependency, 210 KiB of x86 code for a simple hello world program.
- ▶ Used in Numcalc (<http://numcalc.com/>), an online scientific calculator with advanced features.

### NumCalc.com

The Scientific Web Calculator

```
Paste Here   Hex output  Numeric mode
> 1/7.0
0.14285714285714285714285714285714285
> \p 192
> 1/7.0
0.14285714285714285714285714285714285714285714285714285714287
> exp(-I*PI)
-1-5.092611592231400345768126043144482942377765447837240117714e-60*I
> (1+2*I)*(2+3*I)
-4+7*I
> a=[[1,2,3],[2,3,4],[5,6,8]]
[[ 1, 2, 3 ], [ 2, 3, 4 ], [ 5, 6, 8 ] ]
> det(a)
-1
> inverse(a)
[[ 0, -2, 1 ], [ -4, 7, -2 ], [ 3, -4, 1 ] ]
> deriv(1/(1+X)+2/(2+X)+3/(3+X))
(-6*X^4-44*X^3-120*X^2-144*X-66)/(X^6+12*X^5+58*X^4+144*X^3+193*X^2+132*X+36)
>
```



# Another cool project by Fabrice Bellard

Lossless compression of natural language text messages

<https://bellard.org/textsynth/sms.html>

- ▶ Not in Buildroot yet!
- ▶ Allows to compress English text by using the probability of the next word computed by the GPT-2 language model released by OpenAI.
- ▶ Achieves better compression than the best general purpose compressors.
- ▶ Only works on x86 CPUs with AVX2 support.

### 3) Compression results

```
-----
```

File	Model #params	Original size (bytes)	Compr. size (bytes)	Ratio (bpb)	CMIX v18 ratio (bpb)
book1	117M	768771	152283	1.58	1.82
book1	345M	768771	142183	1.48	
book1	774M	768771	137562	1.43	
book1	1558M	768771	134217	1.40	
alice29.txt	117M	152089	23615	1.24	1.65
alice29.txt	345M	152089	20587	1.08	
alice29.txt	774M	152089	19096	1.00	
alice29.txt	1558M	152089	17382	0.91	
enwik5	117M	100000	14875	1.19	1.60
enwik5	345M	100000	13511	1.08	
enwik5	774M	100000	13240	1.06	
enwik5	1558M	100000	12918	1.03	

#### Notes:

- book1 comes from the Calgary corpus.
- alice29.txt comes from the Canterbury corpus.
- enwik5 contains the first 100000 bytes of the English Wikipedia dump of March 3, 2006 (<http://matmahoney.net/dc/textdata.html>).
- For best performance, use the UTF-8 encoding and don't mix CRLF and LF line breaks.
- For reference, the results of CMIX (<http://www.byronknoll.com/cmix.html>) are provided.





Modern tool to assist in network address calculations  
for IPv4 and IPv6

<https://gitlab.com/ipcalc/ipcalc>

- ▶ Added on 2020-10-14
- ▶ A fork of `ipcalc` from Fedora, which has now since replaced it.
- ▶ Easy to use from scripts and also supports JSON output for further processing.

#### IPv4

```
$ ipcalc --all-info 193.92.150.2/24
Address:      193.92.150.2
Network:     193.92.150.0/24
Netmask:     255.255.255.0 = 24
Broadcast:   193.92.150.255
Reverse DNS: 150.92.193.in-addr.arpa.
```

```
Address space: Internet
Address class: Class C
HostMin:      193.92.150.1
HostMax:      193.92.150.254
Hosts/Net:    254
```

```
Country code: GR
Country:      Greece
```

```
$ ipcalc -pnmb --minaddr --maxaddr --geoinfo --addrspace 193.92.150.2/255.255.255.224
NETMASK=255.255.255.224
PREFIX=27
BROADCAST=193.92.150.31
NETWORK=193.92.150.0
MINADDR=193.92.150.1
MAXADDR=193.92.150.30
ADDRSPACE="Internet"
COUNTRY="Greece"
```

Shell script to run simple system-level validation tests on embedded Linux firmware

<https://github.com/savoirfairelinux/cukinia>

- ▶ Added on 2020-05-05
- ▶ Easy to use `/bin/sh` script depending only on BusyBox.
- ▶ Can output results in JUnit XML (for Jenkins and other CI tools) and CSV

```
tests ↵ 6b0a292 ↵ ? ↵ 1 ↵ $ cukinia test1.conf
cukinia: --- passing tests ---
cukinia: [PASS] : Checking user "colord"
cukinia: [PASS] : Checking user "hacker" (is false)
cukinia: [PASS] : Checking process "Xorg" as johndoe
cukinia: [PASS] : Checking process "Xwayland" as any user
cukinia: [PASS] : Checking http url "http://localhost:631/"
cukinia: [PASS] : Checking python package: math
cukinia: [PASS] : Checking mount: nfsd on /proc/fs/nfsd with rw relatime options
cukinia: [PASS] : Running "test -f /etc/passwd"
cukinia: --- includes ---
cukinia: [PASS] : Checking user "root"
cukinia: [PASS] : Checking user "toor" (is false)
cukinia: --- executables ---
cukinia: [PASS] : External: ./tests.d/01-pass
cukinia: [FAIL] : External: ./tests.d/02-fail
cukinia: --- should all fail ---
cukinia: [FAIL] : Checking user "colord" (is false)
cukinia: [FAIL] : Checking user "hacker"
cukinia: [FAIL] : Checking process "Xorg" as johndoe (is false)
cukinia: [FAIL] : Checking process "Xorg" as johndoe2
cukinia: [FAIL] : Checking http url "http://localhost:632/"
cukinia: [FAIL] : Checking mount: /dev/mapper/loop2p2 on /mnt with data=ordered options
cukinia: [FAIL] : Checking python package: meth
cukinia: [FAIL] : Running "test -f /etc/passwd" (is false)
cukinia: [FAIL] : Running "test 4 -gt 10"
cukinia: ran 21 tests, 10 failed
tests ↵ 6b0a292 ↵ ? ↵ 1 ↵ $
```



# htpdate

Synchronize system time with web servers  
as a reference time source

<https://github.com/angeloc/htpdate>

- ▶ Added on 2020-08-07
- ▶ Implementing the HTTP Time Protocol (HTP)
- ▶ Useful when NTP UDP ports are blocked on your network
- ▶ Relies on timestamps in HTTP headers
- ▶ Accuracy in the 0.5 second range. Improves when multiple servers are used.

Usage:

```
htpdate [-046abdh1qstxD] [-i pid file] [-m minpoll] [-M maxpoll]
[-p precision] [-P <proxyserver>[:port]] [-u user[:group]]
<host[:port]> ...
```

Eg. `htpdate -q www.example.com`  
Eg. `htpdate -a -t https://www.example.com http://www.example.com`

In general, if more web servers are specified, the accuracy will increase.

See manpage for more details.

## New features

Version v1.2.4 introduced https support.

## Maintenance

This project is a fork based on htp from Eddy Vervest (<http://www.vervest.org/htp>). This fork exists only because original project is unmaintained and there are no other active forks.





## Multi base interactive calculator

<https://github.com/mellowcandle/bitwise>

- ▶ Added on 2020-07-06
- ▶ Targeting low level hackers, kernel developers and device drivers developers
- ▶ Command line interface for quick operations
- ▶ Allows to toggle bits and apply bitwise operations and shifts in interactive mode.

### Examples:

#### Simple base conversion

```
stdcall@stdcall-pc:~/dev $ bitwise 0x1231230
Decimal: 19875645
HexDecimal: 0x1231230
Octal: 0118611875
Base: 16.19 MiB
ASCII: .....#-#
Binary:
0 0 0 0 0 0 1 | 0 0 1 0 0 0 1 1 | 0 0 0 1 0 0 1 0 | 0 0 1 1 1 0 1
31 - 24      23 - 16      15 - 8       7 - 0
```

#### C style syntax Calculator

```
stdcall@stdcall-pc:~/dev $ bitwise "0x30 * 0x20 + 30 / 2"
Decimal: 1551
HexDecimal: 0x06F
Octal: 03017
Base: 1.51 KiB
ASCII: .....
Binary:
0 0 0 0 1 1 0 | 0 0 0 1 1 1 1
15 - 8       7 - 0
```

#### Interactive mode

*bitwise* starts in interactive mode if no command line parameters are passed or if the `-i` / `--interactive` flag is passed. In this mode, you can input a number and manipulate it and see the other bases change dynamically. It also allows changing individual bits in the binary. You can show the help screen by pressing `F1`.



Access your device's terminal from anywhere via the web.

<https://github.com/zhaojh329/rtty>

- ▶ Added on 2020-03-10
- ▶ Allows to access your device through http (with SSL and TLS)
- ▶ Easy to upload files and download files
- ▶ Alternatives in Buildroot: *ttyd* (added 2020-06-16), *shellinbox* (older)
- ▶ My advice: use a VPN instead, this way your device doesn't need a public IP address and doesn't have a public interface.

```
← → C Not secure | 192.168.10.42:5913/rtty/ABE494191F17
OpenWrt login: root
Password:

BusyBox v1.31.1 () built-in shell (ash)

┌───┴───┐
├───┬───┤
│   │   │
│   │   │
│   │   │
│   │   │
│   │   │
│   │   │
│   │   │
│   │   │
│   │   │
├───┬───┤
└───┴───┘
    W I R E L E S S   F R E E D O M

-----
OpenWrt SNAPSHOT, r14864-c8a81ada58
-----
root@OpenWrt:~# cd /tmp/
root@OpenWrt:/tmp# rtty -R
Waiting to receive. Press Ctrl+C to cancel
Transferring 'macaddress.io-db.json'...
 100%   9.83 MB   2.907s
root@OpenWrt:/tmp# rtty -S macaddress.io-db.json
Transferring 'macaddress.io-db.json'...Press Ctrl+C to cancel
 20%   2.04 MB   0.101s
```

## Out Of Memory killer in userspace

<https://github.com/rfjakob/earlyoom>

- ▶ Added on 2020-06-12
- ▶ Acting earlier than the kernelspace OOM killer, that kicks in when the system is already on its knees.
- ▶ Still killing the same process with the highest `/proc/*/oom_score`

### earlyoom - The Early OOM Daemon

CI passing license MIT release v1.6.2 lgsm alerts 0 code quality: c/c++ A+

The oom-killer generally has a bad reputation among Linux users. This may be part of the reason Linux invokes it only when it has absolutely no other choice. It will swap out the desktop environment, drop the whole page cache and empty every buffer before it will ultimately kill a process. At least that's what I think that it will do. I have yet to be patient enough to wait for it, sitting in front of an unresponsive system.

This made me and other people wonder if the oom-killer could be configured to step in earlier: [reddit r/linux](#), [superuser.com](#), [unix.stackexchange.com](#).

As it turns out, no, it can't. At least using the in-kernel oom-killer. In the user space, however, we can do whatever we want.

#### What does it do

earlyoom checks the amount of available memory and free swap up to 10 times a second (less often if there is a lot of free memory). By default if both are below 10%, it will kill the largest process (highest `oom_score`). The percentage value is configurable via command line arguments.



## Older nuggets (1)

Actively maintained of course

- ▶ *Motion* (2016-09-27)  
<https://motion-project.github.io>  
Motion detector software for video cameras
- ▶ *pv* - Pipe viewer (2011-10-07) - <http://www.ivarch.com/programs/pv.shtml>  
Add progress information to your command line pipes.
- ▶ *sshfs* (2009-11-24) - <https://github.com/libfuse/sshfs>  
Very convenient to share files with a server or embedded system through ssh.  
Great to access build outputs on big servers.
- ▶ *webp* (2013-01-24) - <https://developers.google.com/speed/webp/>  
Superior alternative to PNG and JPEG, can be lossy and lossless
- ▶ *zlog* (2014-03-03) - <https://github.com/HardySimpson/zlog>  
Reliable, high-performance, thread safe, flexible, pure C logging library.





## Older nuggets (2)

- ▶ *Eigen* (2013-09-05) - <http://eigen.tuxfamily.org/>  
C++ vector computation library.  
No compiling necessary, just need to include the headers.
- ▶ *chrony* (2013-05-07) - <https://chrony.tuxfamily.org/>  
Alternative to ntpd, can also use time from GPS frames.
- ▶ *linenoise* (2012-12-05) - <https://github.com/antirez/linenoise>  
A BSD alternative to GNU readline.
- ▶ *msmtp* (2012-04-17) - <https://marlam.de/msmtp/>  
Convenient and lightweight SMTP client to relay mail to an external SMTP server. Caution: GPLv3
- ▶ *mosh* (2015-07-26) - <https://mosh.org>  
SSH client that resists disconnects.



## Older nuggets (3)

- ▶ *ngrep* (2011-09-10) - <https://github.com/jpr5/ngrep/>  
Network grep - Lightweight alternative to tcpdump
- ▶ *nload* (2016-03-08) - <http://www.roland-riegel.de/nload/>  
Console application monitoring network traffic and bandwidth usage in real time (not actively maintained for 4 years)
- ▶ *tini* (2018-08-10) (reverse of "init") - <https://github.com/krallin/tini>  
Simplest init process to avoid zombies. Meant to be used in containers but could be used when you start your application directly (no init program) - Use `init=/tini -- /app` in the kernel command line.
- ▶ *xxhash* (2015-07-16) - <https://github.com/Cyan4973/xxHash>  
Very fast hashing algorithm, running at RAM speed limits.

Questions?  
Suggestions?  
Comments?

Michael Opdenacker  
*michael.opdenacker@bootlin.com*

Slides under CC-BY-SA 3.0  
<https://bootlin.com/pub/conferences/2021/elc>

Please continue to contribute to the public pad:  
<https://annuel2.framapad.org/p/buildroot-nuggets-9pmi?lang=en>