

NAME

lircd - LIRC daemon decodes infrared signals and provides them on a Unix domain socket.

SYNOPSIS

lircd [*options*] [*config-file*]

DESCRIPTION

The main task of lircd is to decode the infrared signals and provide an uniform interface for client applications. Clients can connect to lircd through a Unix domain socket which is located in `var/run/lirc/lircd`. Using this socket they will get the infrared codes received by lircd and they can send commands to lircd.

- h --help**
display this message
- v --version**
display version
- n --nodaemon**
don't fork to background
- p --permission=*mode***
file permissions for `/var/run/lirc/lircd`
- H --driver=*driver***
use given driver
- d --device=*device***
read from given device
- l --listen[=*address*:*port*]**
listen for network connections
- c --connect=*host*[:*port*]**
connect to remote lircd server
- o --output=*socket***
output socket filename
- P --pidfile=*file***
daemon pid file
- L --logfile=*file***
daemon log file
- r --release[=*suffix*]**
auto-generate release events
- a --allow-simulate**
accept SIMULATE command
- u --uinput**
generate Linux input events
- R --repeat-max=*limit***
allow at most this many repeats

OPTIONS

The `--permission` option gives the file permission of `var/run/lirc/lircd` if it has to be created in octal representation. Read the documentation for `chmod` for further details. If no `--permission` option is given when the socket is initially created the default is to give all users read and write permissions (0666 in octal representation). If `/var/run/lirc/lircd` already exists this option has no effect.

With the `--device` option you can select the character device which lircd should read from. The default currently is `/dev/lirc` but it probably will change in future.

If you're using the dev/input driver, you can use *name=STRING* or *phys=STRING* to select the device; lircd will look in /dev/input to find a device with a matching description. This is useful in case the device name isn't fixed. *STRING* may contain the '*' and '?' wildcards and '' to mark them as literal.

With the --listen option you can let lircd listen for network connections on the given address/port. The default address is 0.0.0.0, which means that connections on all network interfaces will be accepted. The default port is 8765. No security checks are currently implemented. The listening lircd instance will send all IR events to the connecting lircd instances.

The --connect option allows you to connect to other lircd servers that provide a network socket at the given host and port number. The number of such connections is currently limited to 100. The connecting lircd instance will receive IR events from the lircd instance it connects to.

With the --output option you can select Unix domain socket, which lircd will write remote key codes to. The default currently is var/run/lirc/lircd.

With the --pidfile option you can select the lircd daemon pid file. The default currently is /var/run/lirc/lircd.pid.

With the --logfile option you can select the lircd daemon log file. The default currently is /var/log/lircd. Note that this option will only be available if you compiled lircd without syslog support.

The --release option enables automatic generation of release events for each button press. lircd will append the given suffix to the button name for each release event. If no suffix is given the default suffix is '_UP'.

The --allow-simulate option will enable the SIMULATE command which can be issued using [irsend\(1\)](#). This will allow simulating arbitrary IR events from the command line. Use this option with caution because it will give all users with access to the lircd socket wide control over your system. E.g. if you have configured your system to shut down by a button press on your remote control, everybody will be able to shut down your system from the command line.

On Linux systems the --uinput option will enable automatic generation of Linux input events. lircd will open /dev/input/uinput and inject key events to the Linux kernel. The key code depends on the name that was given a button in the lircd config file, e.g. if the button is named KEY_1, the '1' key code will be generated. You will find a complete list of possible button names in /usr/include/linux/input.h.

The --repeat-max option sets an upper limit to the number of repeats when sending a signal. The current default is 600. A SEND_START request will repeat the signal this many times. Also, if the number of repeats in a SEND_ONCE request exceeds this number, it will be replaced by this number.

FILES

The config file for lircd is located in /etc/lirc/lircd.conf. lircd has its own log file in /var/log/lircd (beginning with LIRC version 0.6.1 you can configure lircd to use syslogd for log messages; then it depends on your system configuration where log messages will show up). You can make lircd reread its config file and reopen its log file by sending the HUP signal to the program. That way you can rotate old log files.

DAEMONS

lircd and lircmd are daemons. You should start them in some init script depending on your system. There are some example scripts for different distributions in the contrib directory. lircmd has to be started after lircd as it connects to the socket lircd provides.

If you start lircd or lircmd from your shell prompt you will usually get back immediately to the prompt. Often people think that the program has died. But this is not an error. lircd and lircmd are daemons. Daemons always run in background.

SEE ALSO

The documentation for **lirc** is maintained as html pages. They are located under `html/` in the documentation directory.