

Dismiss

Join GitHub today

GitHub is home to over 40 million developers working together to host and review code, manage projects, and build software together.

Sign up

A guide for creating a Quake III Arena dedicated server on Linux.

4 commits

2 branches

0 releases

1 contributor

Branch: master

New pull request

Find file

Clone or download

roguephysicist added two links concerning config files

Latest commit c341210 on Nov 28, 2017

README.md	added two links concerning config files	2 years ago
autoexec.cfg	initial commit	4 years ago
bots.cfg	initial commit	4 years ago
levels.cfg	initial commit	4 years ago
q3start.sh	initial commit	4 years ago
server.cfg	initial commit	4 years ago

README.md

# Installing a Quake III Arena dedicated server

This is a guide for setting up a dedicated Q3A server on Linux. It is mostly distribution-agnostic so it will work on almost any Linux flavor with little or no modification. The required software is available for many hardware platforms, so you can very quickly setup your own server on almost any kind of computer. A Raspberry Pi 2 or 3 makes an excellent Q3A server!

This server will run a specific game type and cycle through maps automatically. If you leave the server running with bots, they will continue to battle it out while you are not connected! This guide has only been tested for the original Q3A, but I imagine that similar steps and configuration files could be used for other games that use this engine.

You can read more about this on the [official ioquake3 documentation](#). See [here](#) and [here](#) for more information about the config files and about maintaining a Q3A server.

You will need:

- Original .pk3 files from the game CD or digital download,
- ioquake3-server , a [modern implementation](#) of the Q3A engine,
- any modern Linux distribution, I use Debian Jessie here.

## Installation

I suggest creating a user with restricted privileges for running the Q3A server. This makes it easier to manage the server. As root ,

```
useradd -m -g users -s /bin/bash -d /home/quake3 quake3 # create quake3 user
passwd quake3 # change quake3's password
```

Install the ioquake3-server package by running

```
apt-get install ioquake3-server # install the Q3A server
```

Switch over to the newly created `quake3` user. The Q3A server installed at `/usr/lib/ioquake3/ioq3ded` , and running it should produce the following:

```
$ /usr/lib/ioquake3/ioq3ded

ioq3 1.36+u20140802+gca9eebb-2+b1/Debian linux-x86_64 Oct 14 2014
Have SSE support
----- FS_Startup -----
Current search path:
/home/quake3/.q3a/baseq3
/usr/lib/ioquake3/baseq3

-----
0 files in pk3 files
"pak0.pk3" is missing. Please copy it from your legitimate Q3 CDRom. Point
Release files are missing. Please re-install the 1.32 point release. Also check
that your ioq3 executable is in the correct place and that every file in the
"baseq3" directory is present and readable
```

We are missing the `.pk3` files from our Q3A CD or digital download.

Running the server produces a `.q3a` directory in the home directory. Copy your Q3A files from the `baseq3` directory here. You can also copy the Quake III: Team Arena files by copying the `missionpack/` directory alongside `baseq3/` , although this is optional for a dedicated server. You should end up with this:

```
.q3a/
├── baseq3
│   ├── pak0.pk3
│   ├── pak1.pk3
│   ├── pak2.pk3
│   ├── pak3.pk3
│   ├── pak4.pk3
│   ├── pak5.pk3
│   ├── pak6.pk3
│   ├── pak7.pk3
│   └── pak8.pk3
└── missionpack
    ├── pak0.pk3
    ├── pak1.pk3
    ├── pak2.pk3
    └── pak3.pk3
```

The server now has everything it needs to run correctly. You can start it up again with `/usr/lib/ioquake3/ioq3ded` , and kill it with `ctrl + c` . Now we just need to configure and automate it a bit.

## Configuration

The server can be configured using `.cfg` files. Typically the only one that is required is `autoexec.cfg` . However, it is very practical to divide the options between different files in order to better manage your server. Here I have divided the options between 4 files each with different functions.

- 1. `autoexec.cfg` controls the most basic aspects of the server,
- 2. `server.cfg` defines the gametype and most game options,
- 3. `bots.cfg` allows for easy setup of the bots playing on the server, and
- 4. `levels.cfg` sets up the maps to be played, their order, and rotation.

I'll list each file here, and they are also included with this guide.

```
$ cat autoexec.cfg

set vm_game 2           // I have no idea what this shit is
set vm_cgame 2          // Nope
set vm_ui 2             // Nada
set dedicated 1         // Dedicated server but not announced
set com_hunkmegs 128    // How much RAM for your server
set net_port 27960      // The network port

$ cat server.cfg
```

```
// general server info
seta sv_hostname "Q3A CTF"           // name that appears in server list
seta g_motd "Hard CTF 24/7"         // message that appears when connecting
seta sv_maxclients 16               // max number of clients than can connect
seta sv_pure 1                      // pure server, no altered pak files
seta g_quadfactor 4                 // quad damage strength (3 is normal)
seta g_friendlyFire 1               // friendly fire motherfucker

// capture the flag
seta g_gametype 4                   // 0:FFA, 1:Tourney, 2:FFA, 3:TD, 4:CTF
seta g_teamAutoJoin 0               // 0:goes into spectator mode, 1:auto joins a team
seta g_teamForceBalance 0           // 0:free selection, 1:forces player on weak team
seta timelimit 30                   // Time limit in minutes
seta capturelimit 8                 // Capture limit for CTF
seta fraglimit 0                    // Frag limit

// team deathmatch
//seta g_gametype 3                  // 0:FFA, 1:Tourney, 2:FFA, 3:TD, 4:CTF
//seta g_teamAutoJoin 0              // 0:goes into spectator mode, 1:auto joins a team
//seta g_teamForceBalance 1          // 0:free selection, 1:forces player on weak team
//seta timelimit 15                  // Time limit in minutes
//seta fraglimit 25                  // Frag limit

// free for all
//seta g_gametype 0                  // 0:FFA, 1:Tourney, 2:FFA, 3:TD, 4:CTF
//seta timelimit 10                  // Time limit in minutes
//seta fraglimit 15                  // Frag limit

seta g_weaponrespawn 2              // weapon respawn in seconds
seta g_inactivity 120               // kick players after being inactive for x seconds
seta g_forcerespawn 0               // player has to press primary button to respawn
seta g_log server.log               // log name
seta logfile 3                      // probably some kind of log verbosity?
seta rconpassword "secret"          // sets RCON password for remote console

seta rate "12400"                   // not sure
seta snaps "40"                     // what this
seta cl_maxpackets "40"             // stuff is
seta cl_packetdup "1"               // all about

$ cat bots.cfg

seta bot_enable 1                   // Allow bots on the server
seta bot_nochat 1                   // Shut those fucking bots up
seta g_spskill 4                    // Default skill of bots [1-5]
seta bot_minplayers 5               // This fills the server with bots to satisfy the minimum

///## Manual adding of bots. syntax:
///## addbot name [skill] [team] [delay]
//addbot doom      4   blue   10
//addbot bones     4   blue   10
//addbot slash     4   blue   10
//addbot orbb      4   red    10
//addbot major     4   red    10
//addbot hunter    4   red    10
//addbot bitterman 4   red    10
//addbot keel      4   red    10

$ cat levels.cfg

set dm1 "map q3ctf4; set nextmap vstr dm2"
set dm2 "map q3ctf3; set nextmap vstr dm3"
set dm3 "map q3ctf2; set nextmap vstr dm4"
set dm4 "map q3ctf1; set nextmap vstr dm1"
vstr dm1
```

These files need to be placed in the `.q3a/baseq3/` directory.

## Running your Q3A server

You can start your server using these config files by running

```
/usr/lib/ioquake3/ioq3ded +exec server.cfg +exec levels.cfg +exec bots.cfg
```

It is possible to run the server in the background or even autostarting it after booting in to the system. However, in this guide I suggest running it manually with the `quake3` user, inside of a GNU Screen session. This makes it very easy to access and monitor. Additionally, the server binary is very verbose and details everything that is occurring in-game. This can be especially amusing when you have bots fighting even when you are not connected.

I have created a simple script that loads the configuration files and starts the server automatically. Remember, the files need to be placed in the `.q3a/baseq3/` directory for the program to see them. You can place the following script directly into the home directory of the `quake3` user:

```
$ cat q3start.sh

#!/bin/bash
# quick starting a quake 3 dedicated server
/usr/lib/ioquake3/ioq3ded +exec server.cfg +exec levels.cfg +exec bots.cfg
```

In summary, you should

1. log on to the computer,
2. start a Screen session with `screen -D -R`,
3. switch to the `quake3` user with `su - quake3`,
4. execute the server with `./q3start.sh`, and
5. enjoy!