

HQmpd – Project Planning

Anders Karlsson Tobias Olausson
andekar@student.chalmers.se *olaussot@student.chalmers.se*

September 7, 2009

Goals

The project is divided into several sub-goals as follows below; the subgoals are merged into the complete planning independently of each other. We have decided on the following sub-goals;

Simple server (15 hours)

The player should support the complete mpd protocol, and have support for multiple clients at once.

Codec (Total: 220 hours)

A decoder for mp3 and/or ogg and/or flac is to be implemented. The code should be written for readability and not for speed.

Output (20-40 hours)

Be able to write uncompressed music to the soundcard in Haskell, for example using OpenAL.

ID3 (10 hours)

Correct understanding and parsing of id3 tags.

Decode (75 hours)

Decode the format(s) into uncompressed format in Haskell.

Control (20 hours)

Forward/backward search in the file.

Optimize (75 hours)

Make the codec run in real-time or faster.

Queue (20 hours)

Extend the mpd protocol to include a queue command, and implement this command in the software.

Simple Client	(Total: 25 hours)
Build a simple client resembling mpc, but with support for the queue command.	
Client API	(15 hours)
Write a Haskell client API for the MPD protocol, including queue support.	
Client Binary	(10 hours)
The actual program that talks to the server.	
Configuration File	(Total: 50 hours)
The file should contain directives to set various server settings.	
.conf	(10 hours)
Like rc.conf or similar, familiar from most GNU/Linux systems.	
Haskell	(20 hours)
Haskell configuration file like in XMonad and Yi, with dynamic recompilation using the dyre library.	
Library	(20 hours)
To make it easier for the user to write own extensions.	
Report writing	(70 hours)
Total	400 hours

Time plan

The project spans over a total of 16 weeks; 36-51. Below follows a table of when things should be started, and when they should be done.

Week	Start	End
36	Introduction, Simple Server	
37	Report, Output	Simple Server
38	ID3, Queue, .conf	ID3
39		Queue, Output
40	Decode	Client API
41		
42	Optimization	Client Binary
43	Control	.conf
44	HaskellConf	Decode
45	Catchup	Report rev 1
46	Catchup	HaskellConf
47	Library,	Control
48	Client API, Client Binary	
49		Library, Report rev 2
50		Client Binary, Client API
51		Optimization, final Report