Project

Haiku (**俳句**) is a very short form of Japanese poetry. One of the main characteristics is that a haiku consist of 17 “on” (*also known as “morae” though often loosely translated as "syllables*"), in three phrases of 5, 7, and 5 on, respectively.

We want to establish a new type of art form, called “Artificial Haiku”, which is based on the same logic, but instead of syllables, we want to use whole words.

For example:
this is the first phrase (5)
second phrase must have exactly seven words (7)
the last one has five (5)

We need a set of programs, which will help us to train new “Artificial Haiku” students to become masters and to keep all this knowledge for the future generations.

1. Haiku Masters Daemon

We need a service, which will control multiple Haiku **Masters**. Each **Master** has a **dictionary** of words, provided as a text file, with one word per line. The **Master** reads words from the dictionary and sends them to all its **Students**. After a **Master** sends a word, there is a **cooldown** period (*defined in seconds*) before it reads the next one. When the dictionary is over, the **Master** starts from the beginning.

The **Masters** should use network sockets to communicate with their **Students**. Each **Master** has its own dedicated **port** on which he listens for new **Students** to connect and then sends the words to them.

The service should allow us to start multiple **Masters** with their specific (*dictionary, cooldown, port*) settings. We should be able to start and stop the whole service, as well as add and remove **Masters** in run-time.

2. Haiku Students Client

The Haiku **Students** listen to the words of one or more **Masters**, collect them and organize them in the proper “Artificial Haiku” form. As soon as a complete line of Haiku (5 or 7 words) is ready, the **Student** should print it on the screen (*collect 5 words and print, collect 7 words and print, collect 5 words and print*). A separating line between the individual Haikus should be printed also.

A **Student** client should be started with **hostname** (*IP*) and list of **ports** (*representing the Masters*). We should be able to **stop** the client at any moment.

3. The Librarian

The role of the **Librarian** program is to collect the complete Haikus produced by the **Students** on the same host system and store them into a common log file. The Haikus in the log file should be clearly separated and identified (*time, Student ID*). The exact format of the file and the way of communication with the **Students** is up to you.