**Администрация**

Тази страница дава приблизително обяснение за администрирането на състезания със spoj0. За да управлявате състезания със spoj0 е необходимо първо да сте инсталирали системата. Необходимо е също така да имате ssh достъп до съответната инстанция (най-често с потребител spoj0, в директорията /home/spoj0).

Препоръчва се да познавате linux командите за управление на файлове и директории, да имате подходящ ssh клиент (ssh или Putty), както и програма за копиране на файлове през ssh (например scp или WinSCP).

**Потребителски действия**

За оставащата част на документа под '.' разбираме директорията /home/spoj0/. Препоръчително е това да бъде текущата директория при повечето действия по управление на системата. В случай на специално инсталиране е възможно тази директория да бъде различна.

Повечето административни задачи от spoj0 се извършват с манипулиране на файловата система и използване на скрипта ./spoj0-control.pl. Темите на състезанията се намират в ./sets/. Препоръчително е скриптът ./spoj0-control.pl да се извиква от неговата директория.

За да получите преглед на командите които предлага скриптът за управление използвайте:

./spoj0-control.pl help

**Пускане и спиране на системата**

Обичайно spoj0 се конфигурира да се пуска автоматично при стартиране на системата. Въпреки това, понякога може да е необходимо да се стартира ръчно. Пускането на системата изисква root привилегии.

Следните команди се използват за пускане и спиране на системата:

| **Скрипт** | **Описание** |
| --- | --- |
| ./spoj0-control.pl start | Стартира системата като системен демон. Изисква root привилегии. |
| ./spoj0-control.pl start-here | Стартира системата на място. Не изисква root привилегии, но остава в текущия терминал. Предвидена е за експериментиране. |
| ./spoj0-control.pl stop | Спира пуснатия демон. Изисква root привилегии. |
| ./spoj0-control.pl kill | Убива пуснатия демон. Изисква root привилегии. Да се използва само в случай на забиване на системата. |

**Добавяне на състезание**

За да добавите състезание в spoj0 е необходимо първо да подготвите състезанието в директория ./sets/<set\_code>/ в необходимия [ManageContestsBg#Формат\_на\_състезанията формат]. След това изпълнявате

./spoj0-control.pl import-set <set\_code>

Ако няма грешки след изпълнение на тази команда, състезанието е качено и готово за работа. Командата извежда и идентфикаторите на качените задачи (problem\_id). Ако състезанието съдържа авторски решения, те се изпращат за оценяване, резултатът от което може да видите в status страницата на системата. Необходимо е да бъдете внимателни при въвеждане на състезания, тъй като някои грешки могат да бъдат оправени след това само с директна манипулация на базата данни.

Имайте предвид, че не трябва да изтривате файловете след качване на състезанието, тъй като оценяващата система чете тестовете от самите тях.

**Обновяване на състезание**

В случай че желаете да промените дадено състезание което вече е качено, може да използвате:

./spoj0-control.pl sync-set <set\_code>

Тази команда може да се използва за:

* промяна на ограниченията на времето за изпълнения на задачите
* промяна на време на започване и продължителност на състезанията
* промяна на името и описанието на състезанието и задачите
* повторно оценяване на авторските решения (в случай че е правена корекция по тях или тестовете)

Тази команда не може да се използва за:

* промяна на кода на състезанието и задачите
* добавяне или премахване на вече добавени задачи и състезания

В случай че е необходима някоя от горните операции, тя трябва да се извърши директно в базата данни.

**Корекция на задачите**

В случай че е необходима промяна по тестовите данни или проверяващата програма (в случай че има такава) е достатъчно директно да се променят необходимите файлове. След като това се направи, е редно да бъдат преоценени и вече оценените решения. За това може да се използва някоя от следните команди:

| **Скрипт** | **Описание** |
| --- | --- |
| ./spoj0-control.pl rejudge-problem <problem\_id> | Маркира всички решения на дадена задача за преоценяване, с изключение на вече приетите за верни такива. |
| ./spoj0-control.pl rejudge-problem-all <problem\_id> | Маркира всички решения на дадена задача за преоценяване, включително вече приетите за верни такива. |
| ./spoj0-control.pl rejudge-run <run\_id> | Маркира точно определено изпратено решение за преоценяване. |

**Други важни неща**

**Използване на килирица**

В описанията на състезанията е напълно допустимо да използвате кирилица. Имайте предвид обаче, че spoj0използва винаги UTF-8 кодировка. За нещастие голяма част от windows системите все още използват кодиране CP1251 поради което е необходим специален софтуер който поддържа UTF-8 (например редакторът Notepad++). Поради възможностите на UTF-8, освен кирилица можете да ползвате и всякакви други азбуки поддържани в Unicode.

**Формат на състезанията**

Всички теми се намират в ./sets/<set\_code>/, където <set\_code> е кодът на съответната тема. Всяка тема представлява директория със следната структура:

* файл с име set-info.conf съдържащ описанието на състезанието (задължително)
* условие на задачите (файл чието име започва с description)
* за всяка задача: директория именувана като кода на задачата

Всяка задача се намира в директория в темата именувана като кода на задачата. Една задача съдържа следните файлове:

* файл с име problem-info.conf съдържащ описанието на задачата (задължително)
* при еднотестови задачи:
  + файл с име test.in съдържащ тестовят входен файл
  + файл с име test.ans съдържащ верният изходен файл
* при многотестови задачи:
  + файлове с име test.00.in, test.01.in, test.02.in и т.н. съдържащи входните файлове
  + файлове с име test.00.ans, test.01.ans, test.02.ans и т.н. съдържащи съответните коректни изходи
  + **внимание:**
    - въпреки че spoj0 поддържа многотестови задачи, към текущият момент системата не поддържа частично оценяване, т.е. задачите се приемат за верни единствено ако минават всички тестове.
    - тестовите двойки трябва да са с последователни номера, започващи от 00, т.е. в тях не трябва да има пропуснати номера.
* изпълним файл с име checker, който представлява проверяваща програма (незадължително)
  + проверяващата програма трябва да приема три аргумента <in\_file> <ans\_file> и <out\_file> които представляват съответно пътищата на тестовия вход, тестовия изход и изхода от изпълнението на решението
  + тя трябва да върне код за грешка 0 ако решението е коректно и код 1 ако не е.
* файлове с имена от вида solution\*.cpp или solution\*.java които представляват авторските решения (незадължително)
* файл с име от вида description\* който съдържа условието на задачата (незадължително, в случай, че няма се търси условие в директорията на темата)

Файлът set-info.conf представлява текстов файл състоящ се от редове от вида <ключ>=<стойност>. За него са дефинирани следните ключове:

* name: име на темата / състезанието
* start\_time: време в което темата / състезанието става активно. Трябва да е коректно написана в ISO 8601 формат **до секундите** включително
* duration: продължителност на темата / състезанието в минути
* show\_sources: 1 или 0. Определя дали след изтичане на времето на състезанието, изпратените решения ще бъдат видими за всички
* about: описание на темата. Може да съдържа HTML.

Пример:

name=DAA 2009 - Practice problems for week 9

start\_time=2009-05-19 08:00:00

duration=2880

show\_sources=1

about=DAA 2009 - Practice problems for week 9 - Dynamic Programming

Файлът problem-info.conf дефинира свойствата на дадена задача. Също представлява текстов файл от ключове и стойности. За него са дефинирани следните ключове:

* name: име на задачата
* time\_limit: цяло число указващо допустимото време за изпълнение в секунди
* about: допълнително описание на задачата. Може да съдържа HTML

Пример:

name=variables

time\_limit=6

about=DAA 2009 - Week 10 - Task 1

**Директна работа с базата данни**

В някои случаи е удачно да се влиза директно в базата данни. Това може да е полезно за преглеждане на подробностите по изпълнение на дадено решение, или за корекция на тема която не може да стане със скрипта за управление.

**Внимание: Директната работа с базата данни изисква познания на SQL и трябва да се прави внимателно, тъй като може да доведе до сериозно повреждане на системата и загуба на данни!**

Можете да влезете в базата на spoj0 с mysql клиента. Например:

mysql spoj0 -u spoj0\_admin -p

Може да използвате стандартни sql заявки. За схемата на базата може да използвате show tables;, describe <име на таблица>, както и sql кода създаващ базата по подразбиране.