

История

Было начало зимы. После праздников, собравшись на работе в офисе, мы обнаружили коробку небольшого размера и черного цвета. На одной из сторон находился небольшой экран, возле которого была надпись "PROJECT-BETA", а на обратной - порт Ethernet-адаптера. Никто не мог вспомнить, как она тут оказалась (праздники ведь были). Первым делом, конечно же, мы сразу подключили ее в свич и увидели, что в сети появилось новое устройство со своим IP-адресом. После сканирования, были обнаружены открытые порты в диапазоне 20015-20115. Подключившись telnet-ом к одному из них, получили некоторую текстовую информацию. И в этот момент нам в офисе отключили свет. Такое иногда бывает, когда много людей на линии греются электрообогревателями. Было темно и поздно и мы пошли домой оставив тогда все как есть.

На следующий день, придя на работу, почувствовали запах горелого в комнате, где находился PROJECT-BETA. Отключив коробку от свитча, обратили внимание на экран. Он был немного поврежденным, но можно было прочитать "GEND P0 1_30 2_1 50_3 P2 1502 1". Пробовали снова подключать коробку в сеть, но результата не было, если не считать процесса нагревания. Оставалось только ее разобрать. Благо возражающих не было.

Внутри мы обнаружили модуль флэш-памяти, а на одном из чипов была надпись PicoJava-IV-Core. Было сделано предположение, что это 100% аппаратная Java-машина. Сделали дампы памяти с флэш-модуля и обнаружили нечто похожее на файлы классов. Попробовали их запустить на JDK 1.6, и ничего не получилось - в файлах были инструкции, которых нет в спецификации JVM. Поковырявшись в этих инструкциях, мы написали модуль загрузки классов (ClassLoader), который модифицирует байт-код перед выполнением.

Как оказалось, "PROJECT-BETA" является игровым сервером, клиентами которого являются устройства с названием Sarka. Что такое Sarka, неизвестно: Google нам не помог. После некоторых экспериментов с запуском сервера стало понятно следующее:

- Каждая Sarka должна иметь уникальное имя.
- Sarka должна быть предварительно сконфигурирована. Данные о конфигурации отправляются на сервер с помощью токенов вида CFG12345678901234567890.
- Токены конфигурации могут быть найдены в памяти сервера, доступ к которой закрыт паролем.
- Прямой доступ к конкретным участкам памяти можно получить с помощью токенов прямого доступа (DMA tokens), например, DMA12345678901234567890.
- Уровень доступа к памяти сервера зависит от текущей конфигурации Sarka и введенных токенов прямого доступа.
- Основное задание Sarka - функционирование в режиме запуска (активизируется командой launch).
- Формат данных, которые принимаются в режиме запуска, зависит от конфигурации Sarka.
- При переходе в режим запуска, Sarka включается в игру. Игра рассчитана на определенное количество игроков и начинается независимо от их подключения, через 5 секунд после запуска сервера "PROJECT-BETA".

- Победитель игры определяется по количеству набранных очков. Информация о правилах начисления очков можно найти в сервере.

Постановка задачи

Ваша задача состоит в программной реализации устройства Sapka и его конфигурации.

PROJECT-BETA лежит тут <ftp://sapka:sapka@stanfy.com.ua/project-beta-1.3.zip>

Еще PROJECT-BETA скоро будет доступен тут: <http://code.google.com/p/sapka-contest-09/downloads/list>

Версия для печати этого задания лежит <ftp://sapka:sapka@stanfy.com.ua/sapka-task-1.3-rus.pdf>

Описание сервера

Для запуска сервера используются скриптовые файлы `start.sh` / `start.bat`.

Они запускают наш модуль загрузки, с именем файла конфигурации в виде параметра. Файл конфигурации (по умолчанию) имеет имя `server.properties`. В папке `lib` лежат необходимые для запуска Java-библиотеки, они подключаются в скриптовых файлах. В папке `maps` лежат файлы, которые используются игровым сервером. Для работы сервера необходимым и достаточным условием является установленный в системе пакет JRE 1.6.

В файле конфигурации могут быть заданы следующие параметры:

```
# Path to the map file
map = maps/simple
```

Префикс имен файлов, которые используются при запуске игрового сервера.

```
# Ports for clients
client.ports = 20015, 20016
```

Список портов, к которым могут подключаться устройства Sapka. Их количество определяет количеством участников в игре.

```
# Trace output file
trace.out = trace.out
```

Имя файла, куда записываются данные о состоянии игры.

```
# Result output file
result.out = result.out
```

Имя файла, куда записываются результаты игры

```
# Visualizer port, it's never waited for
```

```
visualizer.port = 20090
```

Порт для специального устройства Visualizer, которое может подключаться к PROJECT-BETA и , которому отсылаются те же данные, что и в trace.out

```
# Flag to wait for all clients before starting the game  
or not (true = 'yes' | 'true' | 'y')  
client.wait = yes
```

Флаг, который позволяет приостановить старт игры, до момента подключения всех Sapka.

```
# Flag to wait the first visualizer  
visualizer.wait = no
```

Флаг, который позволяет приостановить старт игры, до момента подключения хотя бы одного устройства Visualizer.

Отображение статуса игры

С версии сервера 1.3 добавлено отображение статуса игры на сервере. Для запуска сервера с отображением используйте скрипты keyboard.bat/keyboard.sh. Параметры teamname config-tokens graph-mode-flag keyboards-count.

teamname - Имя команды

config-tokens - токены конфигурации

graph-mode-flag - 'yes', 'y' - включает "красивое" отображение

keyboards count - количество клавиатурных клиентов

Конфигурация сервера при проведении соревнования

Будут использоваться следующие параметры сервера:

```
map = maps/xxx  
client.ports = 20015, 20016 [, 20017, 20018]  
trace.out = trace.out  
result.out = result.out  
visualizer.port = 20090  
client.wait = no //Сервер стартует игру через 5 (пять) секунд после запуска.  
visualizer.wait = no
```

ВАЖНО

При этом не будет возможности использовать DMA-токены, и memconfig.

В качестве уникального имени устройства необходимо использовать имя своей команды, заменив пропуски на знаки подчеркивания или дефисы.

Токены, которые вы будете находить, зависят от указанного имени устройства (teamname), поэтому определитесь с ним сразу же.

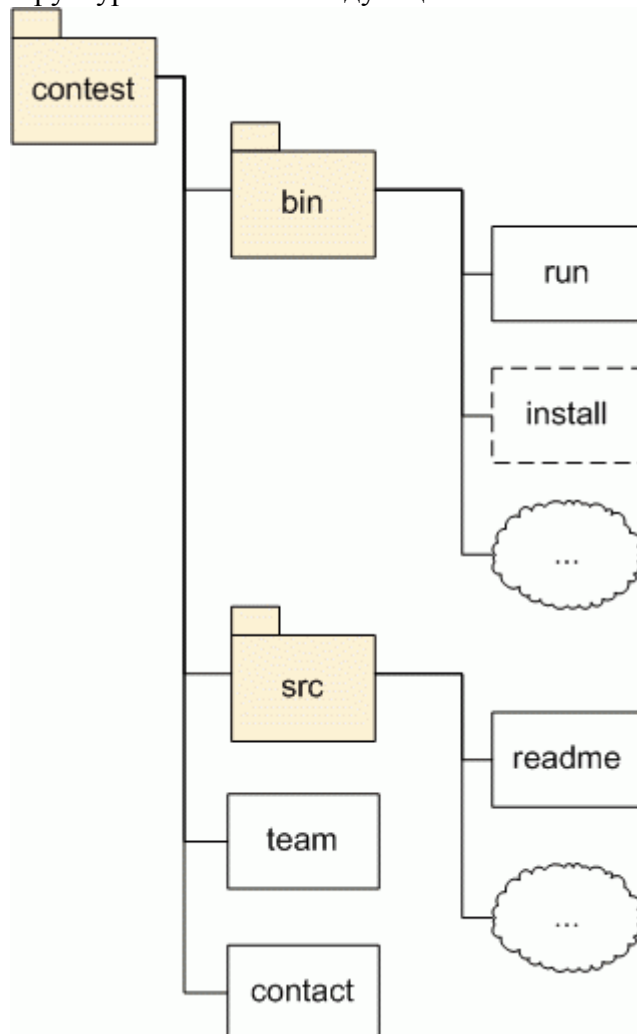
Рекомендованной первой командой при проведении соревнования является

```
launch teamname cfg-token[#cfg-token ...]
```

Она сконфигурирует ваше устройство в соответствии с указанными именем команды и токенами, а сразу же присоединит к игре.

Правила загрузки решений

При первой загрузке вам будет выдан уникальный идентификатор вашей команды, который необходимо использовать для дальнейших загрузок решения. Решения можно перезагружать много раз, но для тестирования будет использоваться последнее загруженное. При загрузке решений, сразу происходит его валидация: проверка на наличие файлов с именем команды, контактов, файла contest/bin/run и на соответствие структуры каталогов следующей схеме.



Описание формата решения:

- решение должно быть запаковано в ZIP-файл
- в корне этого файла должна лежать папка **contest**.
- файл **team** должен содержать название команды (одна строка)
- файл **contact** должен содержать контактные **e-mail** адреса(по одному e-mail в строке в формате Ivan Ivanov <iVanov@gmail.com>). Первая строка файла - основной контакт (капитана) с которым можно связаться в случае чего.
- В папке **bin** должны быть один или два файла - **run** и **install**(**install** - не обязателен), плюс, к этому всему - то, что они будут запускать. То есть ваши бинарники, или интерпретируемые файлы должны лежать тоже в этой папке. Если язык интерпретируемый, то можно все оставить в **src**, как вам больше нравится.
- В папке **src** должны лежать ваши исходники. Кроме того, в файле **readme** можете написать что-нибудь интересное, например какие библиотеки используются и зачем.

Запуск ваших решений будет осуществляться следующим образом:

```
cd contest
if test -r bin/install ; then
chmod +x bin/install
bin/install
fi
chmod +x bin/run
bin/run hostname port
```

Скрипт запуска будет выполняться из-под пользователя user (не root).

Live CD

Для тестирования присланных решений мы будем использовать систему на базе Ubuntu, аналогичную той, что доступна на Live CD. Там у нас есть пользователь stanfy (пароль - также stanfy), под ним, без каких-либо запросов о логине/пароле, и выполняется вход в систему.

Для того, чтобы иметь права суперпользователя, необходимо выполнить

```
sudo -i
```

и, введя пароль stanfy получить рутовые права. Это если вам понадобится настраивать Live CD.

Установлены такие библиотеки

java	Sun JRE	1.6.0_12
c/c++	GCC	4.3.2
haskell	GHC	6.8.2
php		5.2.6
ruby		1.8.7
perl		5.10.0

scheme	MzScheme	4.0
python		2.5.2
c#	Mono	1.9
caml	Ocaml	3.10.2
caml	Camlp4	3.10.2
lisp	Sbcl	1.0.18

Закачать Live CD можно с торрентов - thepiratebay или freexchange.ru

Если с торрентов совсем никак, тогда можете попробовать с нашего сервера <ftp://sapka:sapka@stanfy.com.ua/sapka-cd-0.2.iso> На зарубежный(не украинский) трафик действует ограничение.

Для проверки используйте md5 (5b713e289d058e07f485f5b9ec73ab0c)

На Live CD решение может не компилироваться. Но должно запускаться. Порядок запуска решений смотрите в разделе "Правила загрузки решений".

Настоятельно рекомендуем протестировать ваше решение на Live CD. Как показывает практика - не протестировал в нужном окружении, значит - не работает. Не забывайте об этом.

Раунды и сроки

Решения будут приниматься с 18:00 13-го марта до 18:00 20-го марта. Все присланные решения будут участвовать в Main раунде соревнования, в котором будут определены два победителя - нулевое место и первое. Нулевое место престижней. У программистов все ж начинается с нуля.

Кроме этого будет отдельно проведен Overclock раунд среди решений присланных, в первые три дня соревнования до 18:00 16 марта, где также будут определены два победителя.

Разделение призового фонда между победителями описано в новости про [призовой фонд](#).

Тестирование и результаты

После 18:00 20-го марта мы уходим в тестирование всего того, что вы нам прислали, процесс хоть и автоматизирован, но за ним надо следить.

Через неделю 29 марта на Codecamp'09 в Киеве мы огласим результаты и откроем все на публичный доступ. До этого мы, скорее всего, покажем пару самых интересных выступлений участников.

Если победитель не сможет приехать в Киев на оглашение результатов, то он все равно останется победителем со всеми причитающимися наградами и призами.

Баги, проблемы и коммуникация

В работе черной коробочки иногда встречаются баги (сказывается не родная JVM), и мы за время общения с ней научились их исправлять. Все взаимодействие по таким вопросам будем вести через [issue tracker проекта на google code](#).

Различные вопросы, замечания, предложения и все что придет в голову можно нести в [гугл группу](#).

Оперативное общение всех участников друг с другом будем осуществлять через IRC канал `irc://freenode/sapka`

Сервер: `irc.freenode.net`, Выполнить команду: `/join #sapka`, Кодировка: `cp1251`

Гарантированными каналами являются [issue tracker](#) и [гугл группа](#).

Удачи!