

Západočeská univerzita v plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Semestrální práce
KIV/PD
Web-sériová konzole

Plzeň, 2012
Martin Bydžovský
bydgam@gmail.com
A12N0058P

Zadání

Najít vhodnou implementaci opensource nebo vytvořit WEB aplikaci, která zpřístupní připojenou sériovou konzoli přes WEB rozhraní.

Popis nalezených řešení

Node.js

Nejprve jsem se rozhodl jít cestou (poměrně nových) technologií node.js a socket.io. Node je platforma, která umožní na serveru spustit javascriptový kód, ten je pak interpretován Google V8 enginem (ten používá prohlížeč Chrome na zpracování klientského javascriptu). Pro tuto platformu existuje množství modulů pro práci se soubory, implementace HTTP protokolu nebo např. obsluhu sériového portu.

Socket.io je potom nadstavba nad protokolem HTTP, kde se po standardní výměně request – response naváže (obousměrné a spojové) TCP spojení. Díky tomu odpadá nutnost, aby se klient pořád dokola ptal serveru, jestli není nějaká aktualizace (nová zpráva/příspěvek atd.). Namísto toho server sám může pushovat zprávy o změně stavu klientovi.

Celou aplikaci se mi podařilo naprogramovat a odladit, bohužel jediné, co API node.js neumožňovalo, bylo posílání signálů (resp. klávesových zkratk) – např. CTRL+A, CTRL-Z atd. do cílového zařízení.

Zdrojové soubory tohoto řešení jsou přiloženy k práci. Instalace a spouštění node.js platformy a potřebných modulů je následující:

```
apt-get install python g++ make
wget http://nodejs.org/dist/v0.8.14/node-v0.8.14.tar.gz
tar xvzf node-v0.8.14.tar.gz
cd node-v0.8.14
./configure
make install
cd /home/user/serialport
npm install serialport socket.io
unzip pd_nodjes.zip
node server.js
```

Shellinabox

Protože první řešení nesplňovalo požadavky na 100%, rozhodl jsem se, že se pokusím zprovoznit aplikaci třetí strany Shellinabox. Tato aplikace funguje jako daemon, poslouchá na specifikovaném portu a po připojení zobrazí do webového prohlížeče kompletní terminálové okno (defaultně čeká na přihlášení uživatele a pak otevře jeho shell, případně toto chování lze upravit a rovnou spustit zadaný příkaz). Toto řešení se nakonec ukázalo jako optimální.

Instalace Shellinaboxu probíhá následovně:

```
git clone https://code.google.com/p/shellinabox/
cd shellinabox
./configure
make install
```

Skript otevírající sériové porty

Z koncepce toho, jak je shellinabox naprogramován je třeba pro každý sériový port mít spouštěnou jednu instanci shellinabox daemona. Protože v momentě, kdy se k sériovému portu připojí více lidí, vznikaly by kolize a připojení uživatelé by si navzájem četli části zpráv podle toho, jak by zrovna OS naplánoval který proces. Proto nad každým sériovým portem poslouchá program screen, který funguje jako supervizor a multiplexuje data přijatá ze sériového portu na všechny aktuálně připojené klienty. Příložený sh skript vypadá následovně:

```
#!/bin/bash

killall screen
killall shellinaboxd

screen -dmS session_dev_tty_usb0
screen -S session_dev_tty_usb0 -X stuff "minicom -o\\n"
shellinaboxd -p 4200 -s '/:root:root:/:screen -x session_dev_tty_usb0' &

screen -dmS session_dev_tty_usb1
screen -S session_dev_tty_usb1 -X stuff "minicom -o\\n"
shellinaboxd -p 4300 -s '/:root:root:/:screen -x session_dev_tty_usb1' &
```

První příkaz spustí screenu v detachovaném módu, druhý do ní pošle příkaz, kterým se uvnitř screeny otevře spojení se sériovým portem a následně se spustí shellinabox daemon, který se připojí k této screeně.

Součástí práce je dále i jednoduchý index.html soubor, který zobrazuje klikací mapu a otevírá popup s odpovídajícím shellinaboxem.

Konfigurace shellinaboxu

Na serveru je potřeba nadefinovat v sh skriptu otevření spojení na jednotlivé sériové porty – pro každý port je třeba napsat výše uvedené tři řádky, přičemž vždy se mění:

- 1) Název vytvářené screen session
- 2) Profil minicomu podle sériového portu
- 3) Port, na kterém shellinabox poslouchá

V index.html souboru se potom mění odkazy klikací mapy podle toho, na jakém portu poslouchá jaké zařízení. Pozor, v definici klikací mapy musí být pro každý odkaz definována třída „popup„a jedinečné id prvku. Jeden odkaz klikací mapy potom vypadá následovně:

```
<area id="/dev/tty0" class="popup" ... >
```

Pro spuštění celé aplikace je třeba mít na serveru nainstalovaný Apache (případně jiný web server) na obsluhu příloženého index.html souboru. Do wwwrootu se tedy nahraje index.html a jquery.js. Poté se na serveru spustí (s právy roota) soubor shellinabox.sh.