



**GYMNASIUM JANA KEPLERA**

**DISPLEJ PRO ZOBRAZENÍ SUPLOVÁNÍ**

**MATURITNÍ PRÁCE**

**AUTOR PRÁCE**

**JAN FRYDRYCH**

**VEDOUCÍ PRÁCE**

**EMIL MILER**



## **Zadání maturitní práce z informatiky**

Jméno studenta: **Jan Frydrych**

Školní rok: **2019/2020**

Platnost zadání: **do 30.9.2020**

Třída: **4.A**

### **Zadání práce:** Displej pro zobrazení suplování

Projekt se zabývá výrobou zařízení, které zobrazuje suplování vytažené ze stránek <https://gjk.cz/o-studiu/rozvrh-a-suplovani/suplovani/> a potřebné informace zobrazí na displeji. Zařízení bude spravováno pomocí Ansible, které umožní údržbu zařízení včetně úvodní instalace a nastavení. Zároveň bude po zapnutí schopné fungovat automaticky bez vnějšího zásahu. Projekt by měl být využit jako náhrada za stávající zobrazovač suplování vedle školní nástěnky.

Vedoucí práce: **Emil Miler**

## Abstrakt

Tato maturitní práce se zabývá především výrobou zařízení, které je spravováno pomocí Ansible, kdy po zapnutí je schopno fungovat automaticky bez vnějšího zásahu. Ansible tak automaticky po zapnutí otevře .html soubor do kterého je vepsána stránka <https://gjk.cz/o-studiu/rozvrh-a-suplovani/suplovani/>, která je upravena pomocí skriptu do ní vloženého.

## Klíčová slova

Ansible, skript, HTML, Firefox, Linux, CentOS 8, python3, PIP,

# Úvod

Vybral jsem si jako svou maturitní práci výrobu zařízení, které zobrazuje suplování vedle školní nástěnky. Připadalo mi to jako užitečné zakončení studia na Gymnasiu Jana Keplera a že tam snad něco po sobě i zanechám.

Dostal jsem od školy k dispozici PC Intel NUC Kit na kterém byly nainstalovány nefunkční Windows 7 bez ovladačů, k ovládání myši a klávesnicí. Rovnou jsem tam tedy nainstaloval distribuci Linuxu a to CentOS 8. Kam jsem nainstaloval python3 a rozšířené balíčky, abych mohl nainstalovat Ansible.

## Ansible – *setup.yaml*

Ansible funguje na principu playbooku, kde každý „play“ udělá danou věc. Takže dané zařízení je pak plně automatické. A funguje po nastartování bez vnějšího zásahu.

Okomentované fungování mého playbooku *setup.yaml*:

Nejdříve umožní používat EPEL – což je repozitář, který nám umožní používat veškeré add-ons. V podstatě obsahuje balíčky Fedory pro distribuce Linuxu jako je CentOS.

Ansible pak vytvoří „display user“, který pojmenuje „rozvrh“, kde se to všechno bude odehrávat.

Poté nainstaluje Xorg, což je grafické jádro celého projektu.

Dále pak instaluje Openbox, což je velmi jednoduchý správce oken, který nám umožní pak otevřít Firefox.

Dále Ansible nainstaluje Firefox a povolí „useru“ respektive „rozvrhu“ nainstalovat grafické jádro.

Pak do „useru“ respektive „rozvrhu“ nahraje skript, který se jmenuje *.xinitrc* a říká, jaké prostředí se má startovat, v jaké závislosti na sobě.

Poté tam nahraje skript *dl\_rozvrh.sh*, který nám vytahuje stránku a přidává k ní skript *supl\_add.html*.

Dále nahrává také již zmíněný skript, pojmenovaný *supl\_add.html*.

Pak tam nahraje také konfiguraci daného grafického zobrazení.

Následovně nahrává *rozvrh.service*, který zobrazí rozvrh ve Firefox.

Následovně nasměruje „boot target“ na grafický, aby se zobrazilo to prostředí, které chci, tím je myšleno prostředí ve kterém se zobrazí Firefox se souborem *supl.html*.

A pak spustí daný *rozvrh.service*, který v tom grafickém zobrazení spustí soubor s rozvrhem ve Firefox.

## Skript – *supl\_add.html*

Je to HTML soubor, aby se mi to to přidávalo jako HTML soubor do HTML souboru, proto jsem zvolil HTML. Jednoduše je to v HTML prostě nejjednodušší.

Napsané to je v JavaScriptu, protože mi to přišlo jako nejjednodušší možné řešení, které funguje. Ještě není dopsán, stále na něm pracuji.

Proto je tam druhý skript, který jsem použil ze stránek volně dostupných JavaScriptů a to konkrétně – <http://www.javascriptkit.com/script/cut179.shtml>, který automaticky posouvá stránku pořád dolů a když dosáhne krajní dolní hodnoty soubor znovu načte.

## Skript – *dl\_rozvrh.sh*

Skript napsaný v shellu, který dělá to že bere zdrojový kód stránky <https://gjk.cz/o-studiu/rozvrh-a-suplovani/suplovani/> a vytvoří soubor *supl\_temp.html* do kterého se pak přidá skript a to *supl\_add.html* a pak to vezme celý soubor *supl\_temp.html* a přesune to do souboru *supl.html*. Pak se tento celý skript na hodinu uspí a po hodině se pustí znovu.

```
while true; do
    curl https://gjk.cz/o-studiu/rozvrh-a-suplovani/suplovani/ >
~/supl_temp.html
    cat ~/supl_add.html >> ~/supl_temp.html
    mv ~/supl_temp.html ~/supl.html
    # Firefox pak nasmerovat na file:///.../supl.html - v
/home/rozvrh/.xinitrc prepsat z nastavene stranky na dane umistení supl.html
    sleep 1h
done
```

## *Xinitrc*

Toto je skript, který dělá to, že spouští v daném pořadí dané prostředí, skripty, aplikace a podobně.

Nejdříve spustí openbox, což je správce oken. Pak pustí skript *dl\_rozvrh.sh*, po kterém se hned nemůže iniciovat Firefox, protože by nestihl vytvořit soubor, který Firefox zobrazuje a tak se Firefox spustí, až po uplynutí 60ti vteřin s daným souborem.

```
#!/usr/bin/sh
openbox &
~/dl_rozvrh.sh &
sleep 60s
#firefox https://gjk.cz/o-studiu/rozvrh-a-suplovani/suplovani/
firefox file://$HOME/supl.html
```

## *rozvrh.service*

Poté co máme připojení a zároveň máme nainstalované prostředí to stáhne a zobrazí rozvrh ve Firefoxu.

```
[Unit]
Description=Zobrazí rozvrh ve Firefoxu
After=network.target
```

```
[Service]
Type=simple
ExecStart=/usr/bin/xinit
User=rozvrh
Restart=always
```

```
[Install]
WantedBy=graphical.target
```

## *rc.xml*

To je konfigurační soubor openboxu, který nám umožní nakonfigurovat si zobrazení. Připsal jsem tam na konec, aby se Firefox zobrazil na celou obrazovku, jinak by se zobrazoval pouze jako okno.

V budoucnu bych ještě rád ošetřil přestat zobrazovat myš a vypnout screen timeout,

## *Xwrapper.config*

Textový soubor který označuje, že každý uživatel může spustit X server. Je to workaround pro bezpečnostní opatření, které překázelo. Obcházím tím tak bezpečnostní opatření, které mi překázelo

```
allowed_users = anybody
```

## *Hosts*

Opět textový soubor, který říká, že ansible má provádět všechny úkony zde lokálně na tomto zařízení.

```
display_nuc ansible_connection=local
```

# Výběr technologií

## Xorg

To jest jediné možné použitelné grafické jádro.

## Openbox

Openbox jsem vybral z toho důvodu, že je jednoduchý, není náročný na operační paměť. A bohatě vystačí k realizaci projektu.

## Firefox

Jednoduchý na používání, nezabírá moc místa a rychle a snadno funguje, bez nějakých chyb.

## Požadavky

Distribuce Linuxu CentOS 8.

Nainstalovaný Python3 a jeho správce balíčku PIP.

Ve kterém nainstalovat Ansible.

## Závěr

Závěrem celý projekt je funkční. Funguje přesně podle zadání maturitní práce. Automaticky uloží stránku do souboru, který pak po úpravě skripty zobrazí ve Firefox. Celé je to spravováno a udržováno pomocí Ansible, který funguje bez problémů.

Na jednom skriptu budu nadále pracovat a to v závislosti na rozběhnutí daného webu v normálním stavu. Aktuálně totiž web zobrazuje pouze, že jsou všichni „Dist“, kvůli Covid-19. Takže nemohu pracovat s tím, s čím bych potřeboval.

Zároveň bych chtěl říct, že tento projekt mě posunul v oblasti informačních technologií neskutečně daleko, protože jsem něco na tento princip dělal úplně poprvé. Byla to pro mě velice zajímavá zkušenost, kterou provázelo spoustu problémů ať už v špatně napsaných skriptech nebo v playbooku a tak podobně.

Práci v takovýchto prostředích jsem znal jen a pouze z teoretického hlediska. A otevřelo mi to úplně nový způsob myšlení a nahlížení na věci v IT.

Ve výsledku mi to dalo opravdu hodně a jsem rád, že jsem tento projekt úspěšně dokončil od začátku až do konce. Snad někdy bude úspěšně využit jako náhrada za stávající zobrazovač suplování vedle školní nástěnky.

## Jak to rozchodit

Předpokládám, že pracuji v Linux distribuci CentOS 8.

Zapotřebí je:

Nejprve nainstalovat Python3 a k tomu PIP.

Pak nahrát veškeré soubory z tohoto repozitáře, to jest:

dl\_rozvrh.sh; hosts; rc.xml; README.md; rozvrh.service; setup.yaml;  
supl\_add.html; xinitrc; Xwrapper.config

poté spustit příkaz v terminálu:

```
ansible-playbook -i hosts setup.yaml
```



## Literatura a zdroje

Skript: scroll skript [Online; 18.02.2020]

URL <http://www.javascriptkit.com/script/cut179.shtml>