

2024 Linux w Systemach Wbudowanych – Laboratorium ćw. 1

Student: Piotr Jankiewicz

Treść zadania

Podczas uruchamiania systemu i jego wyłączania, skrypt wysyła e-maila z powiadomieniem o tym fakcie

Procedura odtworzenia projektu z załączonego archiwum

1. Przenieść plik .config do buildroota
2. Przenieść plik skryptu S60startscript z archiwum do ./tmp/init.d
3. Przenieść plik send_email.py z archiwum do ./tmp/init.d
4. Zbudować obraz systemu komendą „make”

Opis rozwiązania

W ramach terminów laboratorium 2 nie udało się wykonać 100% treści zadania. Wymagania zostały spełnione do momentu odpalania skryptu przez system po uruchomieniu, lecz nie wysyła ostatecznie email. Prezentowane rozwiązanie powstało zgodnie z krokami wskazanymi w pliku z przygotowaniem do laboratoriów. Kluczowym w rozwiązaniu okazało się wskazanie foldera overlay, w którym znajdowały się skrypt i program w python3, które były uruchamiane podczas rozpoczęcia systemu v /tmp/init.d i nadanie odpowiedniej nazwy skryptu by był on uwzględniany podczas uruchomienia. Problematycznym okazał się brak menadżera pakietów pip, lecz udało się znaleźć opcję załączania odpowiednich bibliotek w obrazie, bez potrzeby instalowania poprzez skrypt.

Opis modyfikacji i konfiguracji Buildrota i kernela

Konfiguracja BR:

1. Rozpakowanie BR
2. Wybierz zewnętrzny zestaw narzędzi (external toolchain) - (Arm AArch64 12.2.rel1 – wybierany automatycznie).

```
BR2_ARCH_HAS_TOOLCHAIN_BUILDROOT=y  
BR2_ARCH_NEEDS_GCC_AT_LEAST_4_8=y  
BR2_ARCH_NEEDS_GCC_AT_LEAST_4_9=y  
BR2_ARCH_NEEDS_GCC_AT_LEAST_5=y  
BR2_ARCH="aarch64"
```

3. Włącz pamięć podręczną kompilatora - BR2_CCACHE (używając domyślnej lokalizacji /malina/jk/br-ccache – zgodnie z sugestią prowadzącego).

```
BR2_CCACHE_DIR="/malina/jankiewicz/ccache-br"
```

4. Ustaw katalog z nakładkami na system plików - BR2_ROOTFS_OVERLAY na "../overlay" (aby umożliwić dodawanie własnych skryptów).

```
BR2_ROOTFS_OVERLAY
```

5. Wybierz pakiet python3 - BR2_PACKAGE_PYTHON3=y (aby umożliwić uruchamianie skryptów napisanych w Pythonie)

```
BR2_PACKAGE_HOST_PYTHON3=y
```

6. Ustawienie Initramfs jako głównego systemu plików.

```
BR2_TARGET_ROOTFS_INITRAMFS=y
```

7. Ustawienie hasła dla root.

```
BR2_TARGET_ENABLE_ROOT_LOGIN=y  
BR2_TARGET_GENERIC_ROOT_PASSWD="1234"
```

8. Załączenie pakietu dropbear w obrazie, w celu umożliwienia logowania się z sieci.

```
BR2_PACKAGE_DROPBEAR=y  
BR2_PACKAGE_DROPBEAR_CLIENT=y
```

9. Zmiana standardowego powitania systemu na „Welcome Piotr Jankiewicz” przy pomocy ustawienia:

```
BR2_TARGET_GENERIC_ISSUE="Welcome to Piotr Jankiewicz"
```

10. Ustawienie czasu i regionu dzięki pakietowi NTP.

```
BR2_PACKAGE_NTP=y
```

11. Umieszczenie skryptu S60startscript w folderze /etc/init.d

Treść skryptu:

```
#!/bin/sh
```

```
for ((i=1; i<=100; i++)); do  
    echo $i  
done
```

```
# Sprawdzanie, czy Python 3 jest zainstalowany  
if command -v python3 >/dev/null 2>&1; then  
    # Uruchamianie programu Python 3  
    python3 send_email.py  
else  
    echo "Python 3 nie jest zainstalowany. Zainstaluj go, aby kontynuować."  
fi
```

Treść pliku send_email.py

```
import smtplib
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
from google.auth.transport.requests import Request
from google.oauth2.service_account import Credentials

# Ścieżka do pliku credentials.json (pobranego z konsoli Google Cloud Platform)
credentials_path = '/ścieżka/do/credentials.json'

# Adres e-mail odbiorcy
receiver_email = "odbiorca@email.com"

# Temat i treść e-maila
subject = "Testowy e-mail"
body = "To jest treść testowego e-maila wysłanego za pomocą Pythona."

# Konfiguracja wiadomości e-mail
message = MIMEMultipart()
message["To"] = receiver_email
message["Subject"] = subject

# Dodanie treści e-maila
message.attach(MIMEText(body, "plain"))

# Uzyskanie dostępu do konta przy użyciu tokena OAuth
credentials = Credentials.from_service_account_file(credentials_path,
scopes=['https://www.googleapis.com/auth/gmail.send'])
credentials.refresh(Request())

# Ustanowienie połączenia z serwerem SMTP
with smtplib.SMTP("smtp.gmail.com", 587) as server:
    server.starttls()
    server.login("twoj@gmail.com", None)

# Wysłanie e-maila
server.sendmail("twoj@gmail.com", receiver_email, message.as_string())

print("E-mail został wysłany!")
```