

Laboratorul 05. Servicii web

Init

În procesul de boot, după ce kernel-ul s-a inițializat, trebuie pornit ansamblul de procese ce vor rula în user-space pe parcursul sesiunii curente. Pentru a realiza acest lucru, kernel-ul va porni daemon-ul **init**. În cazul în care kernel-ul nu reușește să pornească acest proces, se va produce ceea ce se cheamă "kernel panic".

Init este primul proces pornit, rulează în user-space, are în mod normal PID-ul 1, va fi părintele tuturor celorlalte procese și va adopta orice proces orfan. Pentru a evita posibile accidente, init este implementat astfel încât să ignore majoritatea semnalelor, chiar și pe cele trimise de procese rulate cu privilegii. Singurele semnale pe care acest proces le tratează sunt cele pentru care există handleri instalate explicit. Există totuși distribuții ce conțin astfel de handleri pentru a implementa comportamente asemănătoare cu **halt** sau **shutdown** atunci când init primește **SIGTERM**.

Odată ce a fost pornit procesul init, acesta are rolul ca mai departe să pornească restul proceselor din sesiune. De-a lungul timpului au existat diverse implementări ale acestei operații. Cea mai simplă abordare este cea folosită pe sistemele BSD. Acestea apelează un script ce se află în `/etc/rc` și conține lista programelor ce trebuie pornite. Init nu face altceva decât să ruleze acest script. Alte implementări se bazează pe evenimente asincrone (precum Upstart) sau daemoni specializați meniți să înglobeze și să înlocuiască init-ul clasic și script-urile necesare acestuia (systemd).

CGI

Serviciul pe care îl vom pune la dispoziție pe RaspberryPi în acest laborator este un server web. Serverul web poate genera dinamic pagini. Rădăcina server-ului este `/var/www/`. Aceasta se realizează prin execuția de scripturi aflate în directorul `cgi-bin` din rădăcina serverului, folosindu-se o metodă standard numită Common Gateway Interface. CGI presupune execuția unui script specificat în cererea HTTP (identificat ca orice alt fișier de pe server), care poate primi parametri dintr-un formular web pe stdin și prin variabilele de mediu, respectiv poate genera pagini dinamice scriind pe stdout.

Exemplu:

În directorul `cgi-bin` din root-ul serverului web creăm un fișier executabil numit `1.sh`. Conținutul acestui fișier este de forma:

`1.sh`

```
#!/bin/sh
echo "Content-type: text/html"
echo ""
echo "<HTML><HEAD><TITLE>Sample CGI Output</TITLE></HEAD>"
echo "<BODY><p>"
echo -n "Generated "
date
echo "</p><pre>"
env
echo "</pre>"
echo "</BODY></HTML>"
```

Pentru executare, în browser se scrie `http://<ip>/cgi-bin/1.sh` [`http://%3Cip%3E/cgi-bin/1.sh`] .

Metoda CGI permite execuția oricărui script pe server, putându-se astfel efectua orice operație. În mod normal s-ar restricționa accesul la sistem spre exemplu rulând serverul web sub un utilizator fără privilegii.

Exemplu de cod HTML:

```
<form method="post" action="/cgi-bin/on.sh">
<input type="submit" name="on" value="LED On" />
</form>
```

Chiar dacă scripturile nu generează ieșire, browserul le va interpreta ca downloaduri goale, pentru că a emis o cerere și așteaptă un fișier ca răspuns, mecanismul CGI se desfășoară doar pe server, fără știrea sau interesul browserului. Trebuie ca ele să trimită un răspuns de tip redirect la pagina inițială pentru a obține o interfață rezonabilă. Aceasta se face adăugând în scripturi următorul cod:

```
echo "HTTP/1.0 302 OK"
echo "Location: /"
echo
```

Header-ul de tip Location specifică unde anume să fie redirectat browser-ul. Spre exemplu, dacă aveți o pagină numită *my_index.html* unde vreți ca utilizatorul să fie redirectat, atunci header-ul va fi: "Location: /my_index.html"

Exerciții

- Toate exercițiile se vor rezolva în mașina virtuală QEMU.
- Pentru acest laborator este nevoie să porniți QEMU cu parametri de rețea de bridge. De asemenea, este necesar să puteți să dați *ping* între mașina host și mașina target/QEMU.

0. **Bash** - Creați un script bash în mașina virtuală QEMU care să afișeze data curentă. Acesta trebuie să poată fi executat din orice locație. Hint: */usr/bin/*

1. **HTTP** - Ne propunem în cadrul laboratorului să instalăm un server web pe mașina emulată de Raspberry Pi în QEMU. Instalați pachetele *lighttpd* și *php5-cgi* în QEMU. Dacă nu aveți conexiune la exterior, puteți descărca o imagine care conține aceste pachete de aici

[<https://drive.google.com/file/d/1GjMss6809YZ5polRdQYXIQSx2OdRW2Eg/view?usp=sharing>]. Validați funcționalitatea server-ului web.

- Navigați pe pagina default. O puteți găsi la http://adresa_IP_eth0_qemu/index.lighttpd.html [http://adresa_IP_eth0_qemu/index.lighttpd.html].

2. **CGI** - Creați în *index.html* (în rădăcina ierarhiei web) un formular web cu metoda *POST*, conținând un buton de tip *submit*. Butonul va trebui să execute un script aflat în directorul */var/www/cgi-bin/*, care la rândul lui va apela scriptul creat la exercițiul 0. Scopul este ca la apăsarea butonului să se afișeze data curentă.

3. **PHP** - Afișați într-o pagină web informații de sistem folosind modulul de PHP.

- Adăugați informații despre parametrii pe care îi primește kernel-ul la pornire
- Adăugați informații pe pagină despre device-ul pe care este montat '/'

Puteți verifica dacă modulul de PHP este activat cu:

phpinfo.php

```
<?php
phpinfo();
?>
```

Dacă aveți eroare 403, trebuie să rulați *lighty-enable-mod fastcgi* și să verificați ca pachetul *php-cgi* să fie instalat. Verificați și ca *lighttpd* să fie configurat corect, și dacă nu, modificați configurația conform: <https://www.cyberciti.biz/tips/lighttpd-php-fastcgi-configuration.html> [<https://www.cyberciti.biz/tips/lighttpd-php-fastcgi-configuration.html>].

Dacă aveți eroare 500, verificați ca nu rulați din folderul *cgi-bin*.

Resurse

- Soluție laborator
- Lighttpd + PHP: <https://wiki.ubuntu.com/Lighttpd+PHP> [<https://wiki.ubuntu.com/Lighttpd+PHP>]
- Wheezy package list deprecation: <https://www.raspberrypi.org/forums/viewtopic.php?f=66&t=237469> [<https://www.raspberrypi.org/forums/viewtopic.php?f=66&t=237469>]
- Source list when using apt-get: <https://wiki.debian.org/SourcesList> [<https://wiki.debian.org/SourcesList>]

si/laboratoare/05.txt · Last modified: 2020/11/16 07:08 by dragos_florin.costea