Eloszott épületvezérlési rendszer megvalósítása

Nagy András
Baráth Áron
Kéri Kálmán

© Conet Kft.

2016. november 29.

Tartalom

1 Bevezetés
- Hardware
- Software

2 Magasszintű szolgáltatások
- Alapok
- Rendszer indulás
- Device-ok és service-ek
- Package management
- Daemon-ok
- Adminisztráció

3 Demo board

Tartalom

1 Bevezetés
- Hardware
- Software

2 Magasszintű szolgáltatások
- Alapok
- Rendszer indulás
- Device-ok és service-ek
- Package management
- Daemon-ok
- Adminisztráció

3 Demo board

Motiváció

- Elosztott épületvezérlés építéséhez kerestünk környezetet
- Skálázható legyen
- Szimulálható legyen
- Elosztott legyen
- Nem találtunk számunkra megfelelőt
- Komplex probléma

Komplex Rendszerek

- Rendszerelemek: megfelelő hardware és software összessége
- Hardware:
  - kell vas, amin futnak a programok
  - ki-/bemenetek kezelése (fogyasztók, érzékelők)
- Software: Több szintű programok összesége

Hardware környezet

- Utasítások beolvasása, értelmezése, végrehajtása, ki-/bemenetek
- Nagyon alacsony szint („vas”)
  - Végrehajtó egység (processzor)
  - Memória (RAM, ROM, FLASH)
  - Perifériák
  - Beavatkozók, érzékelők
- Sokféle: komplex, nem nyitott, nem hibamentes
- Nehéz kezelni

Hardware alap

- Végrehajtó egység (processzor)
- Sokféle típus
  - Általános célú (asztali gép, szerver, tablet)
  - Speciális célú (grafika, hang)
  - Mikrokontrollerek
- Elmosódtak a határvonalak
- Technológiai fejlődés hatása
  - 1994: Intel Pentium P54C (75-120 MHz, 8-12W, $849)
  - 2011: LPC1830 ARM Cortex M3 (180 MHz, 0,25W, $5)

Hardware tulajdonságok

- Több félét használunk
- Alacsonyabb szinten Microcontroller (ARM, Cortex M0, M3, M4, M7)
- Magasabb szinten Microprocessor (Intel x86_64, ARM Cortex A8)

Software rétegek

- Több szint, különböző szintek (bonyolultság változik)
- Egymásra épülnek
- Alacsonyabb
  - Másként nehéz, pl. időkritikus
- Magasabb
  - Összetett

Programok software környezete

- Elfedi a hardware környezetet
- Egységes software felületet biztosít
  - Több szintű (pl. interpreter, web böngésző)
- Lehet „meztelen vasra” is programot írni
  - Beágyazott rendszerekben nem ritka
  - Operációs rendszer!

Operációs rendszer

- Alacsony szintű software réteg
- Szolgáltatások egységes felülete
- Széles spektrum
- Két nagy pólus: Microkernel és Monolitikus kernel
- Hol húzzuk meg az operációs rendszer határvonalát
  - Operációs rendszer részei egyben vagy külön
  - Szétosztott vagy központosított
- Sebesség, komplexitás, modularitás vonzata van
- Hibrid kernel (részben micro, részben monolitikus)

Tervezett operációs rendszer

- Microkernel
- Valós idejű ütemezés
- Multitasking, multithreading
- Virtuális memória kezelése, de egy címtér
  - ASLR adja magát
- Task kontextusában futó megszakítások
- Nagyon kevés architektúra függő rész

Microkernel tulajdonságok

- Objektum orientált szemléletmód
- Alapvetően C-ben + C preprocesszált ASM
- Gyors, egyszerű rendszerhívások (timeout)
- Single stack kernel (stateless)
- Paraméter átadás csak regisztereken keresztül
- Üzenet alapú kommunikáció

Alacsony szintű szolgáltatások

- A kernel önmagában kevés
- Szükség van a kernel-re épülő szolgáltatásokra
  - Programok betöltése
  - Device driverek írása
  - Task-ok felügyelete
  - Programkönyvtárak
- Mindent külön programokban kell megoldani

Tartalom

1 Bevezetés
- Hardware
- Software

2 Magasszintű szolgáltatások
- Alapok
- Rendszer indulás
- Device-ok és service-ek
- Package management
- Daemon-ok
- Adminisztráció

3 Demo board

Registry

- Futási idejű információk szálbiztos megosztása
  - task lista
  - device lista
- Értesítési mechanizmus
- Típusok: dir, tmp, int, bin
- Szerializálhatóság
- Guarded key

Library Service

- Program méretét növeli a static library
- Érdemes a programkönyvtárat külön tenni
- Dinamikus hozzászerkesztés: dynamic library
- Shared Executable (sx)
  - Stateless (gyorsabb, olcsóbb)
  - Stateful (univerzálisabb)

bootfs és cpio

- Az adatok bootfs-en tárolódnak
- `bootloader.ini`
- Rendszer image-ek: *.cpio.xz
- Multi-config image
- Tartalma teljes egészében a `/run` alá kerül (tmpfs)
- Alapvető device-ok
  - IDE, SATA vezérlő
  - Network driver
  - VFS, file rendszerek
  - Config-ok, script-ek, devman ini-k

Config script

- Egyszerű szintaxis (értékadás, szekvencia és elágazás)
- Alapműveletek elvégzése egész számokon
- String konkatenáció
- Konverziók
- Programindítás
- Műveletek a registry-vel:
  - kulcs létrehozás
  - kulcsra várakozás
  - kulcs törlés

Device manager

- Supervisor
- A detektorok információi ide futnak be
- Programok (újra)indítása
  - `Watch` objektum szerepe
- Függőségek kezelése
- A futtatási környezet egyszer értékelődik ki
- Config script-ek futtatása
- File rendszer interface: `/dev`

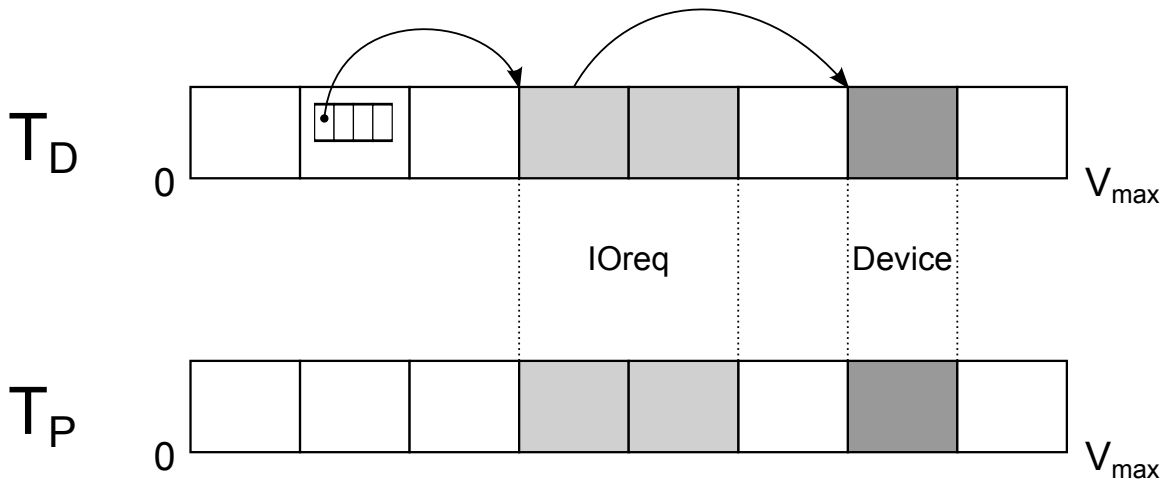
```
/$ ls -l /dev
dr-xr-xr-x     0  2016-11-29  7:05:20  detectors
dr-xr-xr-x     0  2016-11-29  7:05:20  drvroot
crw-rw-rw-     0  2016-11-29  7:05:20  null
dr-xr-xr-x     0  2016-11-29  7:05:20  programs
crw-rw-rw-     0  2016-11-29  7:05:20  random
crw-rw-rw-     0  2016-11-29  7:05:20  zero
```

Detektorok

- Hardware konfiguráció detektálása (ACPI, PCI)
  - Másra is használható
- Statikus (PCI) és dinamikus (USB) detektálások
- A device manager-nek küldi el
- A detektált hardware meghajtója kommunikálhat a detektorával
  - Szabványos interface

I/O request

- Megnyitott Device példány szükséges hozzá
- Nagy mennyiségű adat is átküldhető
- Üzenet küldése a Device-nak
- A rendszer egyik alap építőeleme



Device class

- Azonos szerepkörű device-ok (service-ek) közös interfésze
  - `DCLS_BLOCK`
  - `DCLS_FILESYSTEM`
  - `DCLS_SERVICE`
  - ...
- Egy device (service) több class-t is megvalósíthat
  - pl. TCP/IP device: `DCLS_TCPIP`, `DCLS_CHAR`, `DCLS_SERVICE`
- Vannak opcionális parancsok
  - pl. `FS_CMD_MKDIR`

File system service

- Class: `DCLS_FILESYSTEM`
- Különböző file system-ek közös interfésze
  - tmpfs
  - ffs
  - bootfs
  - FAT32
- Nem feltétlenül valósít meg minden műveletet
- Összekötés blokk vagy flash device-szal
- Mindegyik a VFS-en keresztül érhető el

Network interface

- Ethernet és Wi-Fi
- Adatcsomagok küldése megadott MAC címre
  - A hogyan nem érdekes
- Közös interfész: `DCLS_NETIFACE`
- Speciális műveletek (`NIF_CMD_GETOPT`, `NIF_CMD_SETOPT`)

TCP/IP stack

- Network interfészek felhasználása
- Absztrakciós réteg
- Robosztus, moduláris és testreszabható
- IPv4 üzenetek küldése
  - ICMP
  - UDP
  - TCP
- Alkalmazás szintű protokollok
  - DNS
  - NetBIOS NS
  - DHCP

pkman

- Package manager
- Teljesen C-ben íródott
- `.pkg.cpio.xz` formátum
- Integritás: SHA-256
- Verifikálás: digitális aláírás (RSA-2048)
- Az alpm és a pacman ihlette
- Multi-arch by design

makepk

- pkman-kompatibilis package előállítása
- Különböző sémák
- A package formátum is paraméter
- Feladata:
  - Külső tool-ok meghívása (build)
  - Package leíró (integritás, aláírás)
  - cpio összeállítása
  - Tömörítés

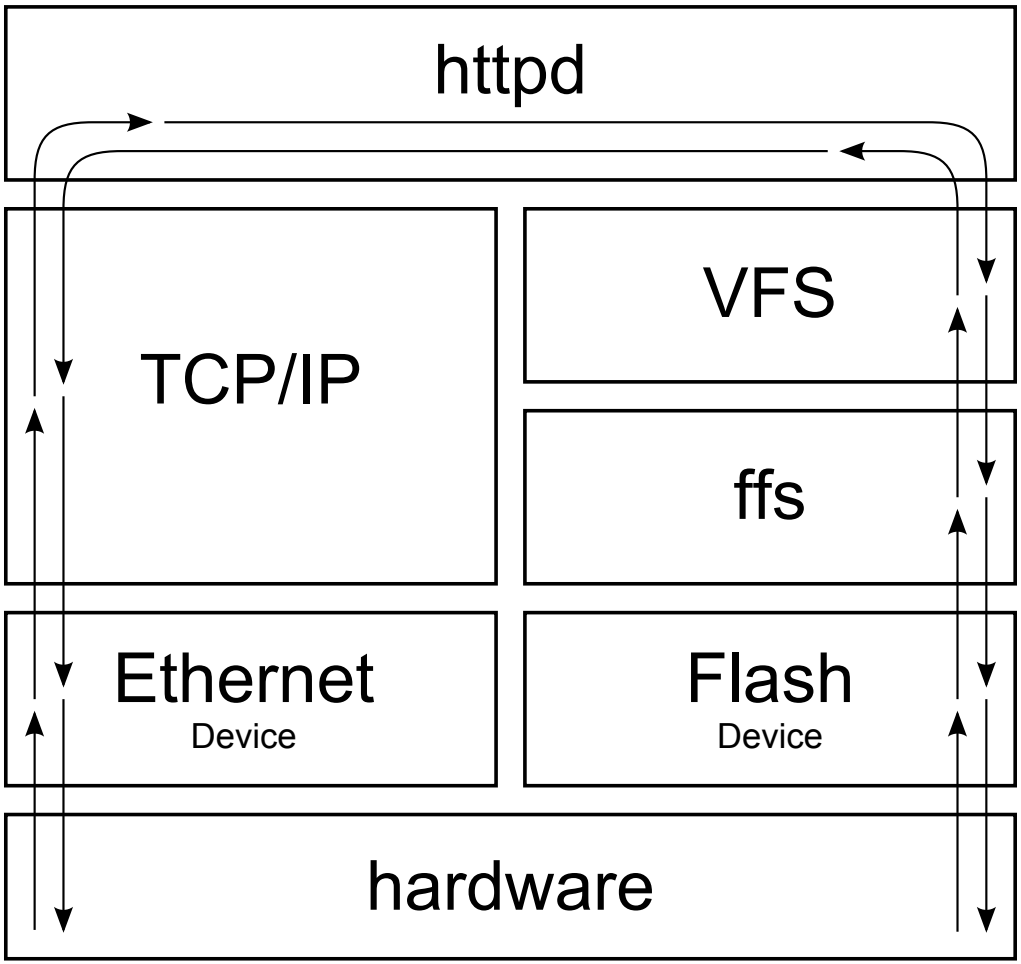
httpd

- Hypertext Transfer Protocol
- FastCGI (és CGI) támogatás
- Többszálú request feldolgozás
- CGI programok automatikus betöltése és cache-elése
  - CGI-k újra felhasználhatók

ftpd

- File Transfer Protocol
- Újra fordított binárisokat gyorsan kellett másolni
  - TFTP kényelmetlen és nem megbízható
- httpd-hez hasonlóan többszálú

HTTP kérés kiszolgálása



barsh

- A bash egy egyszerűsített verziója
- Nincs központi szerepe
- Automatikus- és környezeti változók kezelése
- Shell script lehetőség
- Figyelembe veszi a shebang-et (interpreter választás)
- Minimális a beépített parancskészlet

utibox

- Linux-ból ismert parancsok gyűjteménye
  - `ls`
  - `cp`
  - `mv`
- Egyetlen program, közös függvények miatt
- Symlink-ek választják ki az eszközt
- Program betöltése lustán!

sysbox

- Rendszer erőforrásait és kernel objektumait kezeli
  - `task`
  - `threads`
  - `treereg`
- Egyetlen program, közös függvények miatt
- Symlink-ek választják ki az eszközt
- Program betöltése lustán!

Tartalom

1 Bevezetés
- Hardware
- Software

2 Magasszintű szolgáltatások
- Alapok
- Rendszer indulás
- Device-ok és service-ek
- Package management
- Daemon-ok
- Adminisztráció

3 Demo board

Demo

- Demo board bemutatása

Köszönjük a figyelmet!

andy2@conet.hu
baratharon@caesar.elte.hu
keri.kalman@gmail.com