

Show pagesource

Old revisions

Recent changes

Search

Trace: » lab1 » lab2 » lab3 » lab4 » lab5r » lab6 » lab7 » lab8

## Power Management

### Table of Contents

Power Management
Power states
Tranziții între stări
Exerciții
Dynamic Frequency Scaling
Politică
Guvernator
Exerciții
Documentație suplimentară

### Power states

ATNGW100 suportă două stări de power management.

#### Standby

Această stare oferă puține avantaje ca economie de putere. CPU-ul încă mai este alimentat, deci starea sistemului de operare nu se pierde, sistemul repornește rapid (echivalent ACPI S1)

#### Suspend-to-RAM

Oferă o economie mai mare de putere, toate componentele sunt puse într-o stare low-power (D3), mai puțin memoria, care este pusă în self-refresh mode.

### Tranziții între stări

Tranziție într-o stare low-power se face cu intrarea sysfs /sys/power/state:

- ~ # cat /sys/power/state enumerează stările acceptate
- ~ # echo standby > /sys/power/state intră în starea de standby
- ~ # echo mem > /sys/power/state intră în suspend-to-ram

Suportul pentru power management este în curs de implementare, deci ieșirea din mem nu este disponibilă. Ieșirea automată din starea standby are loc în cazul unei întreruperi (pe serială – apăsarea oricărei taste în terminalul de serială de pe mașina virtuală - sau MMC – cauzată de inserarea unui nou card).

Pentru ca un driver să fie notificat de tranzițiile între stările de power management trebuie ca el să se înregistreze la un notificator de power management.

```
/* kernel/power/main.c */
extern int register_pm_notifier(struct notifier_block *nb);
extern int unregister_pm_notifier(struct notifier_block *nb);

struct notifier_block {
    int (*notifier_call)(struct notifier_block *, unsigned long, void *);
    struct notifier_block *next;
    int priority;
}
```

Apelurile sunt definite în linux/suspend.h și struct notifier\_block este în linux/notifier.h. Notificatorul este implementat cu o listă înălțuită ce ține cont de prioritatea la inserție.

### Exerciții

- Urmăriți cu ajutorul LXR (uitați-vă pe versiunea de kernel 2.6.27.6) implementarea apelului register\_pm\_notifier (din kernel/power/main.c) până ajungeți la adăugarea în listă. Ce valoare trebuie să aibă câmpul priority din notifier\_block pentru ca modulul care se înregistrează să aibă prioritate maxima?
- notifier\_call primește propria celula notifier\_block, împreună cu doi alți parametri, un întreg și un pointer. Urmăriți implementarea și apelurile din kernel/power/main.c ale funcției pm\_notifier\_call\_chain, care apelează pe rând toate funcțiile înregistrate. Cu ce parametri sunt apelate funcțiile?
- Completați scheletul de laborator pentru prima parte (lab8\_1.zip) pentru a crea un modul care postează mesaje de suspend și resume când este notificat de schimbări ale stării de power management.

**\*ATENȚIE\*** Dacă root este montat pe cardul SD, nu o să puteți să îl scoateți pentru a genera întrerupere la inserare! Schimbați întâi root-ul pe partiția de flash!

```
mount -t jffs2 /dev/mtdblock1 /mnt
mount -t proc none /mnt/proc
mount -t sysfs none /mnt/sys
mount -o bind /dev/ mnt/dev
mount -o remount,ro /
mkdir -p /mnt/old_root
pivot_root /mnt /mnt/old_root
```

### Dynamic Frequency Scaling

Kernelul Linux oferă control asupra frecvenței de rulare a procesorului. La început a existat un mecanism simplu de setare statică a unei anumite frecvențe, printr-o interfață /proc. Soluția aceasta nu a fost însă suficientă, o valoare mică a frecvenței oferă un consum redus, limitând puterea de procesare în orice condiții, în timp ce o valoare mare a frecvenței are consumul de energie mare chiar atunci când procesorul este în idle.

Soluția la aceste problemă rezidă într-un mecanism dinamic de modificare a frecvenței, care în Linux este implementată cu guvernatori (eng. governors).

## Politică

Înainte de a defini ce face un guvernator, trebuie mai întâi explicată politica de frecvență a sistemului de operare. Politica presupune limite ale frecvenței, împreună cu un guvernator care decide ce frecvență trebuie folosită. Politica poate fi schimbată pentru fiecare procesor în parte prin interfața sysfs asociată:

```
cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_min_freq
cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq
cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_cur_freq
```

## Guvernator

Guvernatorul este cel care decide care frecvență din plaja pe care o permite politica trebuie folosită. Guvernatorul curent este în `scaling_governor`, iar toți guvernatorii disponibili (pentru care există suport în kernel) sunt în pseudo-fișierul `scaling_available_governors`.

```
cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_available_governors
```

### Tipuri de guvernatori

#### Performance

Unul dintre guvernatorii simpli, care păstrează frecvența la valoarea maximă, oricare ar fi gradul de ocupare al procesorului. Frecvența va fi deci cea mai mică definită de politică, adică `scaling_max_freq`.

Codul poate fi găsit în `drivers/cpufreq/cpufreq_performance.c`

#### Powersave

Opus guvernatorului performance, menține frecvența minimă, `scaling_min_freq`.

Codul poate fi găsit în `drivers/cpufreq/cpufreq_powersave.c`

#### Userspace

Un alt guvernator simplu, care permite utilizatorului să decidă la ce frecvență să lucreze procesorul, prin interfața dată de fișierul `scaling_setspeed`.

Codul poate fi găsit în `drivers/cpufreq/cpufreq_userspace.c`

#### OnDemand

Un guvernator care crește frecvența atunci când gradul de ocupare mediu al procesorului depășește o valoare stabilită și o scade când gradul de ocupare este sub o valoare. Procesorul trebuie să fie capabil de schimbări destul de dese între frecvențe. Interfața acestui guvernator se află în `/sys/devices/system/cpu/cpu0/cpufreq/ondemand`.

Guvernatorul monitorizează la anumite intervale (`sampling_rate` din interfață), verifică dacă nu s-a depășit un prag (`up_threshold`). `Ignore_nice_load` determină guvernatorul să ia sau nu în calcul procesele 'nice' atunci când calculează gradul de ocupare al procesorului. O valoare de 0 va însemna că se pot face calcule lungi cu frecvență mică într-un proces 'nice'.

Codul poate fi găsit în `drivers/cpufreq/cpufreq_ondemand.c`

#### Conservative

Variantă de guvernator asemănătoare cu OnDemand, dar care variază mai puțin brusc frecvența procesorului.

### Notificatoare

Kernel-ul pune la dispoziție notificatoare pentru schimbarea frecvenței procesorului, pentru ca dispozitivele sistemului să poată ține cont de acest parametru al sistemului.

```
int cpufreq_register_notifier(struct notifier_block *nb, unsigned int list);
int cpufreq_unregister_notifier(struct notifier_block *nb, unsigned int list);
```

- `nb` este o structură `notifier_block` cum ați întâlnit la exercițiul anterior
- `list` este lista la care să fie înregistrată funcția:
  - `CPUFREQ_TRANSITION_NOTIFIER` este coada de notificare pentru tranzițiile între frecvențe, vor fi generate evenimente atât înainte, cât și după schimbarea efectivă. Callback-ul va fi apelat fie cu `CPUFREQ_PRECHANGE`, fie cu `CPUFREQ_POSTCHANGE`.
  - `CPUFREQ_POLICY_NOTIFIER` este coada de notificare atât pentru tranziții cât și pentru schimbarea de politici.

Callback-ul va primi în câmpul de date un pointer la o structură de tip `cpufreq_freqs`

## Exerciții

1. Cercetați interfața `cpufreq` din `/sys`
  - Care este frecvența minimă a procesorului AP7000?
  - Care este frecvența maximă a procesorului AP7000?
  - Ce guvernatori sunt configurați?

- Care este guvernatorul implicit?
- 2. Schimbați guvernatorul astfel încât să fie folosită frecvența minimă
- 3. Schimbați guvernatorul astfel încât să fie folosită frecvența maximă
- 4. Schimbați politica astfel încât să fie folosită frecvența minimă
- 5. Schimbați politica astfel încât să fie folosită frecvența maximă
- 6. Cercetați sursele guvernatorilor prezentați.
  - Guvernatorii fac parte din miezul nucleului, sau pot fi folosiți ca module?
  - Ce se întâmplă în powersave atunci când este schimbată politica?
- 7. Pornind de la următorul schelet ( [lab8\\_2.zip](#)), scrieți un modul care să urmărească schimbările de frecvență pe NGW și să aprindă două LED-uri atunci când este folosită frecvența maximă și unul singur atunci când este folosită o frecvență mai mică.

## Documentație suplimentară

---

- Documentație Kernel:
  - Documentation/gpio.txt
  - Documentation/power/states.txt
  - Documentation/cpu-freq/governors.txt
  - Documentation/cpu-freq/user-guide.txt
- sysfs
- ACPI
- Mirror LXR din China (versiune foarte veche de kernel)

si/lab/lab8.txt · Last modified: 2009/12/02 13:35 by Andrei

Show pagesource

Old revisions

Login

Index

Back  
to top

Except where otherwise noted, content on this wiki is licensed under the following license: [CC Attribution-Noncommercial-Share Alike 3.0 Unported](#)



