



Laborator 5

Întreruperi

Sisteme de Operare 2 (SO2)

Departamentul de Calculatoare

Comunicarea cu hardware-ul

Tratarea intreruperilor

Dispozitive des întâlnite

Concluzii

- ▶ porturi I/O
 - ▶ adrese I/O asociate dispozitivelor
 - ▶ pot fi mapate la adrese din memoria fizică
 - ▶ pot fi scrise și/sau citite
 - ▶ sunt structurate în registre specializate
(de obicei: control, stare, intrare, ieșire)
 - ▶ 8, 16 sau 32 biți
- ▶ Întrerupere sau IRQ (Interrupt ReQuest)
 - ▶ modul prin care un dispozitiv semnalează un eveniment
 - ▶ sunt tratate de rutine specializate
 - ▶ rulează într-un context restricționat
 - ▶ pot fi partajate între mai multe dispozitive

- ▶ alocare porturi: `request_region`, `release_region`
- ▶ I/O: `inb`, `outb`, `inw`, `outw`, `inl`, `outl`

Comunicarea cu hardware-ul

Tratarea întreruperilor

Dispozitive des întâlnite

Concluzii

- ▶ request_irq
- ▶ free_irq
- ▶ flag-uri pentru obținere IRQ:
 - ▶ IRQF_SHARED – IRQ partajat de mai multe dispozitive
 - ▶ IRQF_SAMPLE_RANDOM – poate fi folosit ca random seed
 - ▶ IRQF_DISABLED – dezactivat

- ▶ `irqreturn_t my_irq_handler(`
 `int irq_no, void *dev_id);`
- ▶ `irq_no` – numărul intreruperii
- ▶ `dev_id` – pointer date private
- ▶ Întoarce:
 - ▶ `IRQ_HANDLED` – întreruperea a fost tratată
 - ▶ `IRQ_NONE` – întreruperea e pentru alt dispozitiv

1. verificarea originii întreruperii

- ▶ în special pentru întreruperile partajate
- ▶ trebuie verificat dacă dispozitivul care ne interesează este cel care a generat întreruperea
- ▶ multe dispozitive au biți în registrul de stare care indică dacă s-a produs o întrerupere și din ce cauză
- ▶ dacă dispozitivul nu a produs întreruperea, ea trebuie pasată mai departe următorului handler

2. confirmarea întreruperii (acknowledge)

- ▶ de obicei constă în setarea unui bit în registrul de control
- ▶ multe dispozitive nu mai generează alte întreruperi până nu primesc această confirmare

3. procesarea întreruperii

- ▶ citirea datelor de la dispozitiv
- ▶ scrierea datelor către dispozitiv

- ▶ dezactivare / reactivare Întrerupere: `disable_irq`,
`disable_irq_nosync`, `enable_irq`
- ▶ dezactivare / reactivare toate Întreruperile (nerecomandat):
`local_irq_disable`, `local_irq_enable`
- ▶ spinlocks: `spin_lock_irqsave` / `spin_unlock_irqrestore`
(dezactivează Întreruperile / revine la starea inițială);
`spin_lock_irq` / `spin_unlock_irq` (dezactivează /
reactivează Întreruperile necondiționat)

Comunicarea cu hardware-ul

Tratarea Întreruperilor

Dispozitive des întâlnite

Concluzii

- ▶ aka RS-232 aka UART 16550
- ▶ fiecare port serial are o adresă de bază începând cu care sunt mapați registre (ex: 0x3f8) și o întrerupere (ex: IRQ 4)
- ▶ are un set de registre printre care: transmit buffer, receive buffer, interrupt enable (IER), FIFO control (FCR), line status (LSR)
- ▶ parametrii de comunicare: viteză, număr biți de date, paritate, număr biți de oprire, control flux

- ▶ de obicei compatibil cu chipul Intel 8042 (i8042)
- ▶ porturile I/O sunt la adresele 0x60 și 0x64 și folosește IRQ 1
- ▶ generează întreruperi când se schimbă starea tastelor
- ▶ are o serie de registre, printre care STATUS, DATA, CONTROL
- ▶ registrul DATA returnează scancode-ul tastei și un flag (apăsată sau ridicată)

Comunicarea cu hardware-ul

Tratarea Întreruperilor

Dispozitive des Întâlnite

Concluzii

- ▶ comunicarea cu diferitele dispozitive se realizează de multe ori prin porturi I/O
- ▶ porturile I/O sunt structurate în registre (de obicei: control, stare, intrare, ieșire)
- ▶ dispozitivele generează întreruperi pentru a semnala evenimente
- ▶ tratarea întreruperilor trebuie făcută cu mare grijă deoarece:
 - ▶ nu toate operațiile sunt permise în contextul lor
 - ▶ pot apărea probleme de sincronizare cu codul ce rulează în context proces

- ▶ IRQ
- ▶ request_region
- ▶ release_region
- ▶ inb, outb
- ▶ request_irq
- ▶ free_irq
- ▶ spin_lock_irqsave
- ▶ spin_unlock_irqrestore
- ▶ port I/O
- ▶ UART