



Pokročilé operační systémy

Přerušení

Petr Krajča



Katedra informatiky
Univerzita Palackého v Olomouci

- primárně mechanismus umožňující reagovat na asynchronní události
- právě prováděná instrukce je dokončena, předáno řízení obslužné rutině
- eliminace aktivního čekání
- pouze signalizace události
- často nutná kooperace s daným zařízením

klávesnice

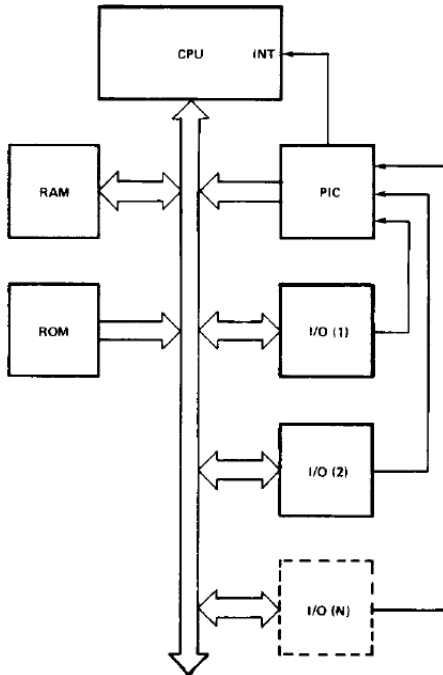
- 1 stisk klávesy vyvolá přerušeni,
- 2 přečtení klávesy přes PMIO/MMIO

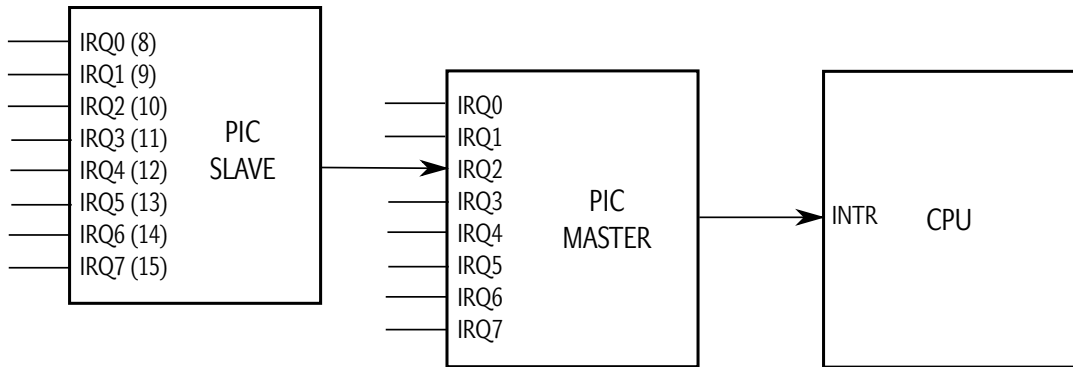
DMA

- 1 konfigurace přenosu (zdroj, cíl, rozsah) přes PMIO/MMIO,
- 2 dokončení přenosu signalizováno přerušeni



- Programmable Interrupt Controller (řídí jednotlivá přerušení, koordinaci, priority)
- dnes pokročilejší řešení (APIC, IOAPIC)
- zapojení (viz bloková schémata)
- kaskáda dvou řadičů přerušení (master/slave)
- programovatelný řadič – různé režimy použití, nutná inicializace (PMIO)







- mechanismus přerušeni plní i další úlohy

Obsluha výjimek (Exceptions)

- *fault* – opravitelná chyba (dělení nulou, výpadek stránky, general protection fault)
- *trap* – vyvolá přerušeni po provedení dané instrukce
- *abort* – vážná systémová chyba (např. neobsloužitelnost výjimky)

Debuggování

- dedikovaná instrukce `int3` (jeden byte)
- trap flag (v registru EF)

Systémová volání

- softwarové přerušeni
- např. Linux, `int 0x80`



- při vzniku přerušení je na zásobník uložena hodnota CS:EIP + EF
- je vyvolána obslužná rutina, adresa uložena v Interrupt Descriptor Table (IDT)
- návrat z obslužné rutiny instrukcí `iret` (obnoví obsah CS:EIP + EF)

Poznámky

- zásobník (zajistit korektní umístění)
- CS:EIP ukazuje na následující instrukci za právě provedenou (až na „fault“, kdy ukazuje na instrukci, která „fault“ vyvolala)
- obvykle nutné zachovat obsah všech registrů (užitečné instrukce `pushad`, `popad`)
- potvrzení o zpracování přerušení
- přerušení lze potlačit pomocí Interrupt Enable Flag (v EF), instrukce `sti` a `cli`
- `interrupt/trap` – nuluje IF/nenuluje IF

- podobný styl práce jako s GDT, jiný typ záznamů
- záznam o velikosti 8 B

bity	význam
0 – 15	spodních 16 b adresy obslužné rutiny
16 – 31	selektor kódového segmentu s obslužnými rutinami
32 – 39	nevyužity (vždy 0)
40 – 43	typ přerušení (16/32, trap, interrupt, ...)
44	0
45, 46	DPL (ochrana přerušení)
47	present – záznam je použit
48 – 63	horních 16 b adresy obslužné rutiny

- adresa IDT uložena v IDTR
- načtení instrukcí lidt, předává se adresa deskriptoru IDT (velikost tabulky - 1 a adresa IDT)

- prvních 32 záznamů vyhrazeno pro výjimky (stavy procesoru)
- obvykle dalších 15 obsluhuje přerušeni z PIC (konfigurovatelné)

záznam	význam	záznam	význam
0	Division by zero	32	IRQ0: Timer
1	Debug	33	IRQ1: Keyboard
2	NMI	34	IRQ2: PIC cascading
3	Breakpoint	38	IRQ6: Floppy
4	Overflow	46	IRQ14: Disk controller
5	Bound Range Exceeded
6	Invalid instruction		
7	No coprocessor		
8	Double fault		
14	Page fault		
...	...		



Asynchronní událost

- Ověřte, že obsluha klávesnice pracuje asynchronně.
- Vypisujte v cyklu postupně na terminál číslo od jedné, dokud nebude stisknuta kombinace kláves Ctrl+C. Použijte obsluhu přerušení.

Obsluha výjimky

- Implementujte vlastní obsluhu výjimky dělení nulou.
- Tato obsluha vypíše text, že došlo k dělení nulou a skončí v nekonečné smyčce.
- Ověřte, že tato obsluha funguje správně. Nezapomeňte, že překladač může zbytečné operace eliminovat.