

Межпроцессорное взаимодействие

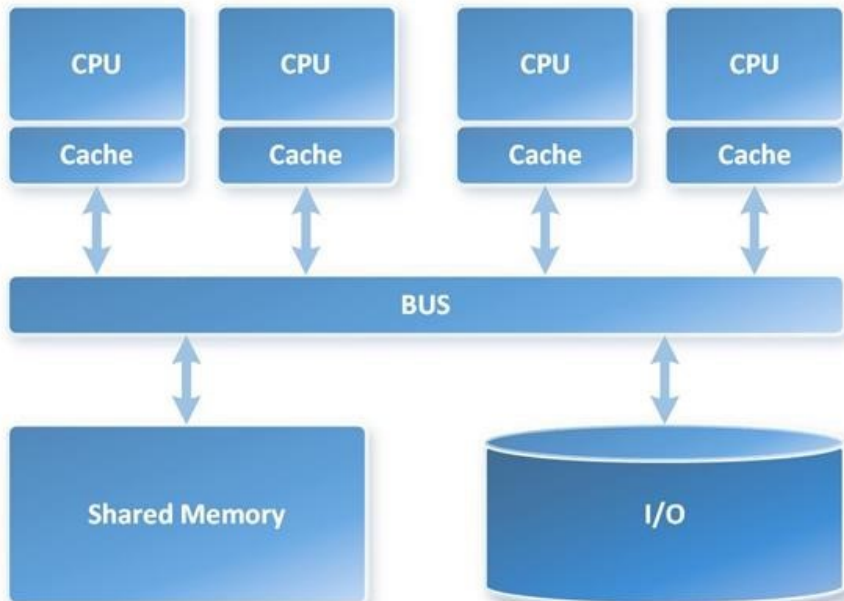
Иван Викторович Михайлов

ИТМО, КТ

imihajlow@gmail.com

25.03.2015

SMP



- Согласованность памяти;
- Согласованность кэша;
- Предсказуемый порядок записей.

Гарантированно атомарные инструкции

- Чтение/запись BYTE;
- Чтение/запись WORD, выровненного на 2 байта;
- Чтение/запись DWORD, выровненного на 4 байта;
- Чтение/запись QWORD, выровненного на 8 байт (64-битный режим);
- Чтение/запись внутри одной линии кэша.

- Сигнал LOCK# – захват шины одним процессором или внешним устройством.
- Префикс LOCK.

С префиксом LOCK можно:

- INC, DEC, NOT, NEG (mem);
- ADD, ADC, SUB, SBB, AND, OR, XOR (mem, src);

XADD

XADD dst, src

$(\text{mem}, \text{src}) \leftarrow (\text{mem} + \text{src}, \text{mem})$

XADD mem, src – можно с LOCK.

BT, BTS, BTR, BTC

BT base, bit

BTS base, bit

BTR base, bit

BTC base, bit

$CF \leftarrow \text{Bit}(\text{base}, \text{bit})$

BTS: $\text{Bit}(\text{base}, \text{bit}) \leftarrow 1$

BTR: $\text{Bit}(\text{base}, \text{bit}) \leftarrow 0$

BTC: $\text{Bit}(\text{base}, \text{bit}) \leftarrow \text{not } \text{Bit}(\text{base}, \text{bit})$

BTS, BTR, BTC в память можно с LOCK.

CMPXCHG

`CMPXCHG dst, src`

```
if EAX == dst:
```

```
    ZF = 1
```

```
    dst = src
```

```
else:
```

```
    ZF = 0
```

```
    EAX = dst
```

`CMPXCHG mem, src` – можно с LOCK.

CMPXCHG8B/CMPXCHG16B

CMPXCHG8B mem64

CMPXCHG16B mem128

CMPXCHG8B dst:

if EDX:EAX == dst:

 ZF = 1

 dst = ECX:EBX

else:

 ZF = 0

 EDX:EAX = dst

CMPXCHG8B mem64 – можно с LOCK.

XCHG mem, src – LOCK по умолчанию.

Mutex

Взаимное исключение – примитив синхронизации, позволяет только одному потоку в один момент времени получить доступ к разделяемым данным.

Spinlock

Реализация взаимного исключения.

Spinlock

locked: dd 0 ; 0 - свободна, 1 - занята

spin_lock:

```
    mov eax, 1
    xchg [locked], eax
    test eax, eax
    jnz spin_lock
    ret
```

spin_unlock:

```
    mov eax, 0
    xchg [locked], eax
    ret
```

PAUSE

PAUSE

Подсказка процессору о том, что это spinlock.

Spinlock

locked: dd 0 ; 0 - свободна, 1 - занята

spin_lock:

 cmp [locked], 1

 jne .get_lock

 pause

 jmp spin_lock

.get_lock:

 mov eax, 1

 xchg [locked], eax

 test eax, eax

 jnz spin_lock

 ret

HLT

Приостановка процессора. Выход из режима остановки:

- Прерывание;
- Отладочное исключение;
- Сброс.

Доступна только на уровне 0.

Spinlock

locked: dd 0 ; 0 - свободна, 1 - занята

spin_lock:

 cmp [locked], 1

 jne .get_lock

 hlt

 jmp spin_lock

.get_lock:

 mov eax, 1

 xchg [locked], eax

 test eax, eax

 jnz spin_lock

 ret

MONITOR/MWAIT

MONITOR – указать адрес для ожидания

- RAX (EAX) – адрес, по которому ждать записей;
- ECX – расширения (*extensions*);
- EDX – подсказки (*hints*).

MWAIT – ждать записи по указанному адресу

- ECX – расширения (*extensions*);
- EAX – подсказки (*hints*).

Доступны на уровне 0, но могут быть и на других.

Spinlock

locked: dd 0 ; 0 - свободна, 1 - занята

spin_lock:

mov eax, locked

monitor

.check_lock:

cmp [locked], 1

jne .get_lock

mwait

jmp check_lock

.get_lock:

mov eax, 1

xchg [locked], eax

test eax, eax

jnz check_lock

ret

CPUID

Получение информации о процессоре.

EAX – параметр.

EAX, EBX, ECX, EDX – результат.

Сериализующая инструкция (все инструкции до нее гарантированно выполнятся).

RTSC

Получение значения счетчика тактов в EDX:EAX.

Если CR4.TSD == 1, то доступна только из уровня 0.

Не является сериализующей.

RTSCP

Получение значения счетчика тактов в EDX:EAX и ID процессора в ECX.

Если CR4.TSD == 1, то доступна только из уровня 0.

Является сериализующей.

RDSEED/RDRAND

RDSEED reg

RDRAND reg

CF = 1, reg = random – успех;

CF = 0, reg = 0 – не получилось.

- Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 3 (3A, 3B & 3C): System Programming Guide (8).

Конец.