1. Describe mutual exclusion, how to implement it? How to implement a mutex? (hits:3)

Peterson's algorithm:

flag[0] = false;

flag[1] = false;

turn;

P0: flag[0] = true;

P0\_gate: turn = 1;

while (flag[1] && turn == 1)

{

// busy wait

}

// critical section

...

// end of critical section

flag[0] = false;

P1: flag[1] = true;

P1\_gate: turn = 0;

while (flag[0] && turn == 0)

{

// busy wait

}

// critical section

...

// end of critical section

flag[1] = false;

Peterson's algorithm for N processes:

// initialization

level[N] = { 0 }; // current level of each process

waiting[N - 1] = { 0 }; // the waiting process of each level 1..N

// code for process #i

for(l = 1; l < N; ++l) {

level[i] = l;

waiting[l] = i;

while((there exists k ≠ i, such that level[k] ≥ l)&& waiting[l] == i)

{

// busy wait

}

}

// critical section

level[i] = 0; // exit section

1. producer/consumer problem. (hits: 4)

Use samephore

1. Design a mutex to ensure bounded waiting.

Add a watching dog thread

1. Deadlock prevention scheme

1) Wait-die: the old waits for the resource hold by young, old wait; young waits for the resource hold by the old, young die;

2) Wound-wait: the old waits for the resource hold by the young, young dies; the young waits for the resource hold by the old, young wait.

1. Make the execution in sequence.

Signal the waiting condition of the next function.

1. What is a volatile variable?

1) Basically you want processor to go and load the value from the memory every time and you don't want the value stored in register.

2) The variable may be changed by some program the compiler does not know.

1. Describe I2S, I2C

1) I2S, also known as Inter-IC Sound, Integrated Interchip Sound, or IIS, is an electrical serial bus interface standard used for connecting digital audio devices. The bus consists of at least three lines: one Bit clock line, one word clock line - also called word select (WS) or left right clock (LRCLK), and one at least one multiplexed data line

2) I²C (Inter-Integrated Circuit, referred to as I-squared-C, I-two-C, or IIC) is a multimaster serial single-ended computer bus.I²C uses only two bidirectional open-drain lines, Serial Data Line (SDA) and Serial Clock (SCL), pulled up with resistors.

1. What is DMA.

1) Memory access without CPU.

2) DMA controller share the system bus with CPU, when DMA takes the bus, CPU is doing some work doesn't require memory access.

3) By using DMA, drivers can access the memory allocated to the user level buffer.

1. Variables in multiple threads

Threads have their own stack but share same heap, static variables on the heap.

1. Spinlock? Various locking mechanisms? How SpinkLocks work in SMP and UP architectures.

A spinlock is a lock which causes a thread trying to acquire it to simply wait in a loop ("spin") while repeatedly checking if the lock is available. Since the thread remains active but is not performing a useful task, the use of such a lock is a kind of busy waiting. Once acquired, spinlocks will usually be held until they are explicitly released, although in some implementations they may be automatically released if the thread being waited on (that which holds the lock) blocks, or "goes to sleep".

Spinlock, mutex, critical section

Symmetric multiprocessing (SMP) involves a multiprocessor computer hardware and software architecture where two or more identical processors connect to a single, shared main memory, have full access to all I/O devices, and are controlled by a single OS instance that treats all processors equally, reserving none for special purposes.

On single-processor systems, spinlocks are not needed because spinlock synchronization is required on high IRQLs only. On high IRQLs (above dispatch IRQL) a context switch cannot occur, so instead of spinning the acquiring thread can simply request an interrupt on the relevant IRQL and return; the interrupt will be masked until the releasing thread lowers the IRQL below the requested IRQL.

1. Difference between semaphore and mutex
2. Say there is an operation like a=b; Its the statement for which you want to avoid concurrency. (Say SMP or preemption or whatever) without using locks (spin, semaphore, mutex etc. etc.) how would you make this statement protected?

Ans: atomic operations (atomic assign etc.)

1. Memory mapped I/O and Input mapped I/O

Memory-mapped I/O uses the same address bus to address both memory and I/O devices the memory and registers of the I/O devices are mapped to (associated with) address values.

Port-mapped I/O often uses a special class of CPU instructions specifically for performing I/O.

1) Also, I/O operations can slow memory access if the address and data buses are shared.

2) One merit of memory-mapped I/O is that, by discarding the extra complexity that port I/O brings, a CPU requires less internal logic and is thus cheaper, faster, easier to build, consumes less power and can be physically smaller; this follows the basic tenets of reduced instruction set computing, and is also advantageous in embedded systems.