本笔记来自kernel 3.14的经验（2014,mv3dp LSP project）

不应该再使用单独的变量jiffies,应为它只是真正的64-bit变量的低32位！

wrong code:

```
early_print("%s-%d\n", __func__, __LINE);
unsigned long temp_1 = jiffies;
early_print("%s-%d-%d\n", __func__, __LINE, temp_1);

while(jiffies < (temp_1 + 10))
{
    early_print("%s-%d-%d-%d\n", __func__, __LINE, jiffies, temp_1 + 10);
}
early_print("%s-%d-%d\n", __func__, __LINE, jiffies);
```

The output:

rest_init-398

rest_init-408-29992

rest_init-408-29991-29982

rest_init-408-29991-29982

.....
rest_init-408-29990-29982

.....

rest_init-408-29989-29982

输出完全出人意料！无法理解。

correct code:

```
u64 temp_1 = get_jiffies_64();
early_print("%s-%d-%llu\n", __func__, __LINE__, temp_1);

while(get_jiffies_64() < (temp_1 + 10))
{
    early_print("%s-%d-%llu-%llu\n", __func__, __LINE__, get_jiffies_64(), temp_1
 + 10);
}
early_print("%s-%d-%llu\n", __func__, __LINE__, get_jiffies_64());
```

Question：在boot阶段，get_jiffies_64()就会返回一个很大的值(4294937304)？

Answer:

in kernel/timer.c

u64 jiffies_64 = INITIAL_JIFFIES;

INITIAL_JIFFIES = -300 * HZ

“Have the 32 bit jiffies value wrap 5 minutes after boot so jiffies wrap bugs show up earlier”

4294937304 = 0xffff,8ad8

选一个大值（接近32-bit的极限0xffff,ffff），是为了使得还是认为jiffies为32-bit的code能尽快出现bug。

Question：在wrong code中输出是jiffies值是递减的?

29992

29991

29990

29989

Answer: unsigned long被当作long来解释。

early_print("%s-%d-**%d-%d**\n", __func__, __LINE, jiffies, temp_1 + 10);

==>

early_print("%s-%d-**%llu-%llu**\n", __func__, __LINE__, get_jiffies_64(), temp_1 + 10);