in lib/Kconfig

```
config ARCH_HAS_SG_CHAIN
        def_bool n
```

CONFIG_ARCH_HAS_SG_CHAIN=y

表示scatterlist支持chain

in include/linux/linux/scatterlist.h

```
static inline void sg_chain(struct scatterlist *prv, unsigned int prv_nents,
                            struct scatterlist *sgl)
{
#ifndef CONFIG_ARCH_HAS_SG_CHAIN
        BUG();
#endif

        /*
         * offset and length are unused for chain entry.  Clear them.
         */
        prv[prv_nents - 1].offset = 0;
        prv[prv_nents - 1].length = 0;

        /*
         * Set lowest bit to indicate a link pointer, and make sure to clear
         * the termination bit if it happens to be set.
         */
        prv[prv_nents - 1].page_link = ((unsigned long) sgl | 0x01) & ~0x02;
}
```

如果CONFIG_ARCH_HAS_SG_CHAIN=n，则invoke sg_chain()完全非法！

```
/**
 * sg_last - return the last scatterlist entry in a list
 * @sgl:        First entry in the scatterlist
 * @nents:      Number of entries in the scatterlist
 *
 * Description:
 *   Should only be used casually, it (currently) scans the entire list
 *   to get the last entry.
 *
 *   Note that the @sgl@ pointer passed in need not be the first one,
 *   the important bit is that @nents@ denotes the number of entries that
 *   exist from @sgl@.
 *
 **/
struct scatterlist *sg_last(struct scatterlist *sgl, unsigned int nents)
{
#ifndef CONFIG_ARCH_HAS_SG_CHAIN
        struct scatterlist *ret = &sgl[nents - 1];
#else
        struct scatterlist *sg, *ret = NULL;
        unsigned int i;

        for_each_sg(sgl, sg, nents, i)
                ret = sg;

#endif
#ifdef CONFIG_DEBUG_SG
        BUG_ON(sgl[0].sg_magic != SG_MAGIC);
        BUG_ON(!sg_is_last(ret));
#endif
        return ret;
}
```

如果不支持ARCH_HAS_SG_CHAIN,则sgl就是一个array,获取最后一个entry就非常简单；但如果ARCH_HAS_SG_CHAIN enable,则sgl完全可能是多个sub-array chain而成，所以只能enumerate而获得最后一个entry。