The genalloc/genpool subsystem

There are a number of memory-allocation subsystems in the kernel, each aimed at a specific need. Sometimes, however, a kernel developer needs to implement a new allocator for a specific range of special-purpose memory; often that memory is located on a device somewhere. The author of the driver for that device can certainly write a little allocator to get the job done, but that is the way to fill the kernel with dozens of poorly tested allocators. Back in 2005, Jes Sorensen lifted one of those allocators from the sym53c8xx_2 driver and posted it as a generic module for the creation of ad hoc memory allocators. This code was merged for the 2.6.13 release; it has been modified considerably since then.

Code using this allocator should include <linux/genalloc.h>. The action begins with the creation of a pool using one of:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\core-api\[linux-master][Documentation][core-api]genalloc.rst, line 20)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: lib/genalloc.c
   :functions: gen_pool_create
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\core-api\[linux-master][Documentation][core-api]genalloc.rst, line 23)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: lib/genalloc.c
   :functions: devm_gen_pool_create
```

A call to gen_pool_create() will create a pool. The granularity of allocations is set with min_alloc_order; it is a log-base-2 number like those used by the page allocator, but it refers to bytes rather than pages. So, if min_alloc_order is passed as 3, then all allocations will be a multiple of eight bytes. Increasing min_alloc_order decreases the memory required to track the memory in the pool. The nid parameter specifies which NUMA node should be used for the allocation of the housekeeping structures; it can be -1 if the caller doesn't care.

The "managed" interface devm_gen_pool_create() ties the pool to a specific device. Among other things, it will automatically clean up the pool when the given device is destroyed.

A pool is shut down with:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\core-api\[linux-master][Documentation][core-api]genalloc.rst, line 41)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: lib/genalloc.c
   :functions: gen_pool_destroy
```

It's worth noting that, if there are still allocations outstanding from the given pool, this function will take the rather extreme step of invoking BUG(), crashing the entire system. You have been warned.

A freshly created pool has no memory to allocate. It is fairly useless in that state, so one of the first orders of business is usually to add memory to the pool. That can be done with one of:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\core-api\[linux-master][Documentation][core-api]genalloc.rst, line 52)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: include/linux/genalloc.h
   :functions: gen_pool_add
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\core-api\[linux-master][Documentation][core-api]genalloc.rst, line

55)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: lib/genalloc.c
   :functions: gen_pool_add_owner
```

A call to gen_pool_add() will place the size bytes of memory starting at addr (in the kernel's virtual address space) into the given pool, once again using nid as the node ID for ancillary memory allocations. The gen_pool_add_virt() variant associates an explicit physical address with the memory; this is only necessary if the pool will be used for DMA allocations.

The functions for allocating memory from the pool (and putting it back) are:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\core-api\[linux-master][Documentation][core-api]genalloc.rst, line 68)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: include/linux/genalloc.h
   :functions: gen_pool_alloc
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\core-api\[linux-master][Documentation][core-api]genalloc.rst, line 71)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: lib/genalloc.c
   :functions: gen_pool_dma_alloc
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\core-api\[linux-master][Documentation][core-api]genalloc.rst, line 74)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: lib/genalloc.c
   :functions: gen_pool_free_owner
```

As one would expect, gen_pool_alloc() will allocate size< bytes from the given pool. The gen_pool_dma_alloc() variant allocates memory for use with DMA operations, returning the associated physical address in the space pointed to by dma. This will only work if the memory was added with gen_pool_add_virt(). Note that this function departs from the usual genpool pattern of using unsigned long values to represent kernel addresses; it returns a void * instead.

That all seems relatively simple; indeed, some developers clearly found it to be too simple. After all, the interface above provides no control over how the allocation functions choose which specific piece of memory to return. If that sort of control is needed, the following functions will be of interest:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\core-api\[linux-master][Documentation][core-api]genalloc.rst, line 91)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: lib/genalloc.c
   :functions: gen_pool_alloc_algo_owner
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\core-api\[linux-master][Documentation][core-api]genalloc.rst, line 94)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: lib/genalloc.c
   :functions: gen_pool_set_algo
```

Allocations with gen_pool_alloc_algo() specify an algorithm to be used to choose the memory to be allocated; the default algorithm can be set with gen_pool_set_algo(). The data value is passed to the algorithm; most ignore it, but it is occasionally needed. One can, naturally, write a special-purpose algorithm, but there is a fair set already available:

- gen_pool_first_fit is a simple first-fit allocator; this is the default algorithm if none other has been specified.
- gen_pool_first_fit_align forces the allocation to have a specific alignment (passed via data in a genpool_data_align structure).
- gen_pool_first_fit_order_align aligns the allocation to the order of the size. A 60-byte allocation will thus be 64-byte aligned, for example.
- gen_pool_best_fit, as one would expect, is a simple best-fit allocator.
- gen_pool_fixed_alloc allocates at a specific offset (passed in a genpool_data_fixed structure via the data parameter) within the pool. If the indicated memory is not available the allocation fails.

There is a handful of other functions, mostly for purposes like querying the space available in the pool or iterating through chunks of memory. Most users, however, should not need much beyond what has been described above. With luck, wider awareness of this module will help to prevent the writing of special-purpose memory allocators in the future.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\core-api\[linux-master][Documentation][core-api]genalloc.rst, line 125)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: lib/genalloc.c
   :functions: gen_pool_virt_to_phys
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\core-api\[linux-master][Documentation][core-api]genalloc.rst, line 128)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: lib/genalloc.c
   :functions: gen_pool_for_each_chunk
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\core-api\[linux-master][Documentation][core-api]genalloc.rst, line 131)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: lib/genalloc.c
   :functions: gen_pool_has_addr
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\core-api\[linux-master][Documentation][core-api]genalloc.rst, line 134)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: lib/genalloc.c
   :functions: gen_pool_avail
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\core-api\[linux-master][Documentation][core-api]genalloc.rst, line 137)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: lib/genalloc.c
   :functions: gen_pool_size
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\core-api\[linux-master][Documentation][core-api]genalloc.rst, line 140)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: lib/genalloc.c
   :functions: gen_pool_get
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\linux-master\Documentation\core-api\[linux-master][Documentation][core-api]genalloc.rst, line 143)

Unknown directive type "kernel-doc".

```
.. kernel-doc:: lib/genalloc.c
   :functions: of_gen_pool_get
```