Subject: [PATCH 0/6][NET-2.6.24] Consolidate private allocations in seq files
Posted by Pavel Emelianov on Tue, 09 Oct 2007 15:50:41 GMT
View Forum Message <> Reply to Message

Many (very many) seq files in net/ allocate some private data
to use it later (mostly for iteration state). All this code
was obviously get using copy-paste method, so move it into
one place.

Almost all of these places either set this private to 0, or
keep uninitialized. Some places, however, pre-initialize this
area, but there are few of them.

The seq_open_private() call just opens the seq file with
allocated and set to zero area. The __seq_open_private() call
makes the same, but returns the allocated memory to the called
to be initialized later.

I didn't measure how much of the .text section this saves,
but I suspect a lot of :) As far as the code is concerned,
this set saves ~450 lines.

Such thing may be useful for any subsystem, but I found this
mostly in the networking code and fixed only it (for a while).

Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

Subject: [PATCH 1/6][NET-2.6.24] Introduce the seq_open_private()
Posted by Pavel Emelianov on Tue, 09 Oct 2007 15:52:58 GMT
View Forum Message <> Reply to Message

This function allocates the zeroed chunk of memory and
call seq_open(). The __seq_open_private() helper returns
the allocated memory to make it possible for the caller
to initialize it.

Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

---

diff --git a/fs/seq_file.c b/fs/seq_file.c
index bbb19be..ca71c11 100644
--- a/fs/seq_file.c
+++ b/fs/seq_file.c
@@ -429,6 +429,39 @@ int seq_release_private(struct inode *in
 }
 EXPORT_SYMBOL(seq_release_private);

```
+void *__seq_open_private(struct file *f, const struct seq_operations *ops,
+  int psize)
+{
+ int rc;
+ void *private;
+ struct seq_file *seq;
+
+ private = kzalloc(psize, GFP_KERNEL);
+ if (private == NULL)
+  goto out;
+
+ rc = seq_open(f, ops);
+ if (rc < 0)
+  goto out_free;
+
+ seq = f->private_data;
+ seq->private = private;
+ return private;
+
+out_free:
+ kfree(private);
+out:
+ return NULL;
+}
+EXPORT_SYMBOL(__seq_open_private);
+
+int seq_open_private(struct file *filp, const struct seq_operations *ops,
+  int psize)
+{
+ return __seq_open_private(filp, ops, psize) ? 0 : -ENOMEM;
+}
+EXPORT_SYMBOL(seq_open_private);
+
 int seq_putc(struct seq_file *m, char c)
 {
  if (m->count < m->size) {
diff --git a/include/linux/seq_file.h b/include/linux/seq_file.h
index 83783ab..8bf1e05 100644
--- a/include/linux/seq_file.h
+++ b/include/linux/seq_file.h
@@ -46,6 +46,8 @@ int seq_path(struct seq_file *, struct v

 int single_open(struct file *, int (*)(struct seq_file *, void *), void *);
 int single_release(struct inode *, struct file *);
+void *__seq_open_private(struct file *, const struct seq_operations *, int);
+int seq_open_private(struct file *, const struct seq_operations *, int);
 int seq_release_private(struct inode *, struct file *);
```

#define SEQ_START_TOKEN ((void *)1)

Subject: [PATCH 2/6][NET-2.6.24] Make core networking code use seq_open_private
Posted by Pavel Emelianov on Tue, 09 Oct 2007 15:55:28 GMT
View Forum Message <> Reply to Message

This concerns the ipv4 and ipv6 code mostly, but also the netlink
and unix sockets.

The netlink code is an example of how to use the __seq_open_private()
call - it saves the net namespace on this private.

Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

```
diff --git a/net/ipv4/arp.c b/net/ipv4/arp.c
index d824819..36d6798 100644
--- a/net/ipv4/arp.c
+++ b/net/ipv4/arp.c
@@ -1378,24 +1378,8 @@ static const struct seq_operations arp_s

 static int arp_seq_open(struct inode *inode, struct file *file)
 {
- struct seq_file *seq;
- int rc = -ENOMEM;
- struct neigh_seq_state *s = kzalloc(sizeof(*s), GFP_KERNEL);
-
- if (!s)
-  goto out;
-
- rc = seq_open(file, &arp_seq_ops);
- if (rc)
-  goto out_kfree;
-
- seq      = file->private_data;
- seq->private = s;
-out:
- return rc;
-out_kfree:
- kfree(s);
- goto out;
+ return seq_open_private(file, &arp_seq_ops,
+   sizeof(struct neigh_seq_state));
 }
```

```
 static const struct file_operations arp_seq_fops = {
diff --git a/net/ipv4/fib_hash.c b/net/ipv4/fib_hash.c
index 9fafbee..527a6e0 100644
--- a/net/ipv4/fib_hash.c
+++ b/net/ipv4/fib_hash.c
@@ -1039,24 +1039,8 @@ static const struct seq_operations fib_s

 static int fib_seq_open(struct inode *inode, struct file *file)
 {
- struct seq_file *seq;
- int rc = -ENOMEM;
- struct fib_iter_state *s = kzalloc(sizeof(*s), GFP_KERNEL);
-
- if (!s)
-  goto out;
-
- rc = seq_open(file, &fib_seq_ops);
- if (rc)
-  goto out_kfree;
-
- seq      = file->private_data;
- seq->private = s;
-out:
- return rc;
-out_kfree:
- kfree(s);
- goto out;
+ return seq_open_private(file, &fib_seq_ops,
+    sizeof(struct fib_iter_state));
 }

 static const struct file_operations fib_seq_fops = {
diff --git a/net/ipv4/fib_trie.c b/net/ipv4/fib_trie.c
index be34bd5..81a8285 100644
--- a/net/ipv4/fib_trie.c
+++ b/net/ipv4/fib_trie.c
@@ -2379,25 +2379,8 @@ static const struct seq_operations fib_t

 static int fib_trie_seq_open(struct inode *inode, struct file *file)
 {
- struct seq_file *seq;
- int rc = -ENOMEM;
- struct fib_trie_iter *s = kmalloc(sizeof(*s), GFP_KERNEL);
-
- if (!s)
-  goto out;
-
```

```
- rc = seq_open(file, &fib_trie_seq_ops);
- if (rc)
-  goto out_kfree;
-
- seq      = file->private_data;
- seq->private = s;
- memset(s, 0, sizeof(*s));
-out:
- return rc;
-out_kfree:
- kfree(s);
- goto out;
+ return seq_open_private(file, &fib_trie_seq_ops,
+   sizeof(struct fib_trie_iter));
 }

 static const struct file_operations fib_trie_fops = {
@@ -2500,25 +2483,8 @@ static const struct seq_operations fib_r

 static int fib_route_seq_open(struct inode *inode, struct file *file)
 {
- struct seq_file *seq;
- int rc = -ENOMEM;
- struct fib_trie_iter *s = kmalloc(sizeof(*s), GFP_KERNEL);
-
- if (!s)
-  goto out;
-
- rc = seq_open(file, &fib_route_seq_ops);
- if (rc)
-  goto out_kfree;
-
- seq      = file->private_data;
- seq->private = s;
- memset(s, 0, sizeof(*s));
-out:
- return rc;
-out_kfree:
- kfree(s);
- goto out;
+ return seq_open_private(file, &fib_route_seq_ops,
+   sizeof(struct fib_trie_iter));
 }

 static const struct file_operations fib_route_fops = {
diff --git a/net/ipv4/igmp.c b/net/ipv4/igmp.c
index 2b6e59c..7dbc282 100644
--- a/net/ipv4/igmp.c
```

```
+++ b/net/ipv4/igmp.c
@@ -2410,23 +2410,8 @@ static const struct seq_operations igmp_

 static int igmp_mc_seq_open(struct inode *inode, struct file *file)
 {
- struct seq_file *seq;
- int rc = -ENOMEM;
- struct igmp_mc_iter_state *s = kzalloc(sizeof(*s), GFP_KERNEL);
-
- if (!s)
-  goto out;
- rc = seq_open(file, &igmp_mc_seq_ops);
- if (rc)
-  goto out_kfree;
-
- seq = file->private_data;
- seq->private = s;
-out:
- return rc;
-out_kfree:
- kfree(s);
- goto out;
+ return seq_open_private(file, &igmp_mc_seq_ops,
+    sizeof(struct igmp_mc_iter_state));
 }

 static const struct file_operations igmp_mc_seq_fops = {
@@ -2584,23 +2569,8 @@ static const struct seq_operations igmp_

 static int igmp_mcf_seq_open(struct inode *inode, struct file *file)
 {
- struct seq_file *seq;
- int rc = -ENOMEM;
- struct igmp_mcf_iter_state *s = kzalloc(sizeof(*s), GFP_KERNEL);
-
- if (!s)
-  goto out;
- rc = seq_open(file, &igmp_mcf_seq_ops);
- if (rc)
-  goto out_kfree;
-
- seq = file->private_data;
- seq->private = s;
-out:
- return rc;
-out_kfree:
- kfree(s);
- goto out;
```

```
+ return seq_open_private(file, &igmp_mcf_seq_ops,
+   sizeof(struct igmp_mcf_iter_state));
 }

 static const struct file_operations igmp_mcf_seq_fops = {
diff --git a/net/ipv4/ipmr.c b/net/ipv4/ipmr.c
index b8b4b49..37bb497 100644
--- a/net/ipv4/ipmr.c
+++ b/net/ipv4/ipmr.c
@@ -1714,26 +1714,8 @@ static const struct seq_operations ipmr_

 static int ipmr_vif_open(struct inode *inode, struct file *file)
 {
- struct seq_file *seq;
- int rc = -ENOMEM;
- struct ipmr_vif_iter *s = kmalloc(sizeof(*s), GFP_KERNEL);
-
- if (!s)
-  goto out;
-
- rc = seq_open(file, &ipmr_vif_seq_ops);
- if (rc)
-  goto out_kfree;
-
- s->ct = 0;
- seq = file->private_data;
- seq->private = s;
-out:
- return rc;
-out_kfree:
- kfree(s);
- goto out;
-
+ return seq_open_private(file, &ipmr_vif_seq_ops,
+   sizeof(struct ipmr_vif_iter));
 }

 static const struct file_operations ipmr_vif_fops = {
@@ -1877,25 +1859,8 @@ static const struct seq_operations ipmr_

 static int ipmr_mfc_open(struct inode *inode, struct file *file)
 {
- struct seq_file *seq;
- int rc = -ENOMEM;
- struct ipmr_mfc_iter *s = kmalloc(sizeof(*s), GFP_KERNEL);
-
- if (!s)
-  goto out;
```

```
-
- rc = seq_open(file, &ipmr_mfc_seq_ops);
- if (rc)
-  goto out_kfree;
-
- seq = file->private_data;
- seq->private = s;
-out:
- return rc;
-out_kfree:
- kfree(s);
- goto out;
-
+ return seq_open_private(file, &ipmr_mfc_seq_ops,
+   sizeof(struct ipmr_mfc_iter));
 }

 static const struct file_operations ipmr_mfc_fops = {
diff --git a/net/ipv4/ipvs/ip_vs_ctl.c b/net/ipv4/ipvs/ip_vs_ctl.c
index 61d023d..7345fc2 100644
--- a/net/ipv4/ipvs/ip_vs_ctl.c
+++ b/net/ipv4/ipvs/ip_vs_ctl.c
@@ -1792,24 +1792,8 @@ static const struct seq_operations ip_vs

 static int ip_vs_info_open(struct inode *inode, struct file *file)
 {
- struct seq_file *seq;
- int rc = -ENOMEM;
- struct ip_vs_iter *s = kzalloc(sizeof(*s), GFP_KERNEL);
-
- if (!s)
-  goto out;
-
- rc = seq_open(file, &ip_vs_info_seq_ops);
- if (rc)
-  goto out_kfree;
-
- seq      = file->private_data;
- seq->private = s;
-out:
- return rc;
-out_kfree:
- kfree(s);
- goto out;
+ return seq_open_private(file, &ip_vs_info_seq_ops,
+   sizeof(struct ip_vs_iter));
 }
```

```
 static const struct file_operations ip_vs_info_fops = {
diff --git a/net/ipv4/raw.c b/net/ipv4/raw.c
index 07070c7..3916fac 100644
--- a/net/ipv4/raw.c
+++ b/net/ipv4/raw.c
@@ -902,24 +902,8 @@ static const struct seq_operations raw_s

 static int raw_seq_open(struct inode *inode, struct file *file)
 {
- struct seq_file *seq;
- int rc = -ENOMEM;
- struct raw_iter_state *s;
-
- s = kzalloc(sizeof(*s), GFP_KERNEL);
- if (!s)
-  goto out;
- rc = seq_open(file, &raw_seq_ops);
- if (rc)
-  goto out_kfree;
-
- seq = file->private_data;
- seq->private = s;
-out:
- return rc;
-out_kfree:
- kfree(s);
- goto out;
+ return seq_open_private(file, &raw_seq_ops,
+   sizeof(struct raw_iter_state));
 }

 static const struct file_operations raw_seq_fops = {
diff --git a/net/ipv4/route.c b/net/ipv4/route.c
index 307e1f1..21b12de 100644
--- a/net/ipv4/route.c
+++ b/net/ipv4/route.c
@@ -375,23 +375,8 @@ static const struct seq_operations rt_ca

 static int rt_cache_seq_open(struct inode *inode, struct file *file)
 {
- struct seq_file *seq;
- int rc = -ENOMEM;
- struct rt_cache_iter_state *s;
-
- s = kzalloc(sizeof(*s), GFP_KERNEL);
- if (!s)
-  goto out;
- rc = seq_open(file, &rt_cache_seq_ops);
```

```
- if (rc)
-  goto out_kfree;
- seq          = file->private_data;
- seq->private = s;
-out:
- return rc;
-out_kfree:
- kfree(s);
- goto out;
+ return seq_open_private(file, &rt_cache_seq_ops,
+    sizeof(struct rt_cache_iter_state));
 }

 static const struct file_operations rt_cache_seq_fops = {
diff --git a/net/ipv6/addrconf.c b/net/ipv6/addrconf.c
index 6d5c3c2..8b2d760 100644
--- a/net/ipv6/addrconf.c
+++ b/net/ipv6/addrconf.c
@@ -2806,24 +2806,8 @@ static const struct seq_operations if6_s

 static int if6_seq_open(struct inode *inode, struct file *file)
 {
- struct seq_file *seq;
- int rc = -ENOMEM;
- struct if6_iter_state *s = kzalloc(sizeof(*s), GFP_KERNEL);
-
- if (!s)
-  goto out;
-
- rc = seq_open(file, &if6_seq_ops);
- if (rc)
-  goto out_kfree;
-
- seq = file->private_data;
- seq->private = s;
-out:
- return rc;
-out_kfree:
- kfree(s);
- goto out;
+ return seq_open_private(file, &if6_seq_ops,
+    sizeof(struct if6_iter_state));
 }

 static const struct file_operations if6_fops = {
diff --git a/net/ipv6/anycast.c b/net/ipv6/anycast.c
index 5810852..f915c4d 100644
--- a/net/ipv6/anycast.c
```

```
+++ b/net/ipv6/anycast.c
@@ -549,24 +549,8 @@ static const struct seq_operations ac6_s

 static int ac6_seq_open(struct inode *inode, struct file *file)
 {
- struct seq_file *seq;
- int rc = -ENOMEM;
- struct ac6_iter_state *s = kzalloc(sizeof(*s), GFP_KERNEL);
-
- if (!s)
-  goto out;
-
- rc = seq_open(file, &ac6_seq_ops);
- if (rc)
-  goto out_kfree;
-
- seq = file->private_data;
- seq->private = s;
-out:
- return rc;
-out_kfree:
- kfree(s);
- goto out;
+ return seq_open_private(file, &ac6_seq_ops,
+    sizeof(struct ac6_iter_state));
 }

 static const struct file_operations ac6_seq_fops = {
diff --git a/net/ipv6/ip6_flowlabel.c b/net/ipv6/ip6_flowlabel.c
index 1791399..217d60f 100644
--- a/net/ipv6/ip6_flowlabel.c
+++ b/net/ipv6/ip6_flowlabel.c
@@ -658,24 +658,8 @@ static const struct seq_operations ip6fl

 static int ip6fl_seq_open(struct inode *inode, struct file *file)
 {
- struct seq_file *seq;
- int rc = -ENOMEM;
- struct ip6fl_iter_state *s = kzalloc(sizeof(*s), GFP_KERNEL);
-
- if (!s)
-  goto out;
-
- rc = seq_open(file, &ip6fl_seq_ops);
- if (rc)
-  goto out_kfree;
-
- seq = file->private_data;
```

```
- seq->private = s;
-out:
- return rc;
-out_kfree:
- kfree(s);
- goto out;
+ return seq_open_private(file, &ip6fl_seq_ops,
+   sizeof(struct ip6fl_iter_state));
 }

 static const struct file_operations ip6fl_seq_fops = {
diff --git a/net/ipv6/mcast.c b/net/ipv6/mcast.c
index cc8d4e2..331d728 100644
--- a/net/ipv6/mcast.c
+++ b/net/ipv6/mcast.c
@@ -2426,24 +2426,8 @@ static const struct seq_operations igmp6

 static int igmp6_mc_seq_open(struct inode *inode, struct file *file)
 {
- struct seq_file *seq;
- int rc = -ENOMEM;
- struct igmp6_mc_iter_state *s = kzalloc(sizeof(*s), GFP_KERNEL);
-
- if (!s)
-  goto out;
-
- rc = seq_open(file, &igmp6_mc_seq_ops);
- if (rc)
-  goto out_kfree;
-
- seq = file->private_data;
- seq->private = s;
-out:
- return rc;
-out_kfree:
- kfree(s);
- goto out;
+ return seq_open_private(file, &igmp6_mc_seq_ops,
+   sizeof(struct igmp6_mc_iter_state));
 }

 static const struct file_operations igmp6_mc_seq_fops = {
@@ -2600,24 +2584,8 @@ static const struct seq_operations igmp6

 static int igmp6_mcf_seq_open(struct inode *inode, struct file *file)
 {
- struct seq_file *seq;
- int rc = -ENOMEM;
```

```
- struct igmp6_mcf_iter_state *s = kzalloc(sizeof(*s), GFP_KERNEL);
-
- if (!s)
-  goto out;
-
- rc = seq_open(file, &igmp6_mcf_seq_ops);
- if (rc)
-  goto out_kfree;
-
- seq = file->private_data;
- seq->private = s;
-out:
- return rc;
-out_kfree:
- kfree(s);
- goto out;
+ return seq_open_private(file, &igmp6_mcf_seq_ops,
+    sizeof(struct igmp6_mcf_iter_state));
 }

 static const struct file_operations igmp6_mcf_seq_fops = {
diff --git a/net/ipv6/raw.c b/net/ipv6/raw.c
index bdd0974..ca24ef1 100644
--- a/net/ipv6/raw.c
+++ b/net/ipv6/raw.c
@@ -1289,21 +1289,8 @@ static const struct seq_operations raw6_

 static int raw6_seq_open(struct inode *inode, struct file *file)
 {
- struct seq_file *seq;
- int rc = -ENOMEM;
- struct raw6_iter_state *s = kzalloc(sizeof(*s), GFP_KERNEL);
- if (!s)
-  goto out;
- rc = seq_open(file, &raw6_seq_ops);
- if (rc)
-  goto out_kfree;
- seq = file->private_data;
- seq->private = s;
-out:
- return rc;
-out_kfree:
- kfree(s);
- goto out;
+ return seq_open_private(file, &raw6_seq_ops,
+    sizeof(struct raw6_iter_state));
 }
```

```
 static const struct file_operations raw6_seq_fops = {
diff --git a/net/netlink/af_netlink.c b/net/netlink/af_netlink.c
index 3ef3282..f934f54 100644
--- a/net/netlink/af_netlink.c
+++ b/net/netlink/af_netlink.c
@@ -1845,27 +1845,18 @@ static const struct seq_operations netli

 static int netlink_seq_open(struct inode *inode, struct file *file)
 {
- struct seq_file *seq;
  struct nl_seq_iter *iter;
- int err;

- iter = kzalloc(sizeof(*iter), GFP_KERNEL);
+ iter = __seq_open_private(file, &netlink_seq_ops, sizeof(*iter));
  if (!iter)
   return -ENOMEM;

- err = seq_open(file, &netlink_seq_ops);
- if (err) {
-  kfree(iter);
-  return err;
- }
-
- seq = file->private_data;
- seq->private = iter;
  iter->net = get_proc_net(inode);
  if (!iter->net) {
   seq_release_private(inode, file);
   return -ENXIO;
  }
+
  return 0;
 }

diff --git a/net/unix/af_unix.c b/net/unix/af_unix.c
index 10e7312..2b57eaf 100644
--- a/net/unix/af_unix.c
+++ b/net/unix/af_unix.c
@@ -2086,25 +2086,7 @@ static const struct seq_operations unix_

 static int unix_seq_open(struct inode *inode, struct file *file)
 {
- struct seq_file *seq;
- int rc = -ENOMEM;
- int *iter = kmalloc(sizeof(int), GFP_KERNEL);
-
- if (!iter)
```

```
-  goto out;
-
- rc = seq_open(file, &unix_seq_ops);
- if (rc)
-  goto out_kfree;
-
- seq      = file->private_data;
- seq->private = iter;
- *iter = 0;
-out:
- return rc;
-out_kfree:
- kfree(iter);
- goto out;
+ return seq_open_private(file, &unix_seq_ops, sizeof(int));
 }

 static const struct file_operations unix_seq_fops = {
```

Subject: [PATCH 3/6][NET-2.6.24] Make netfilter code use the seq_open_private
Posted by Pavel Emelianov on Tue, 09 Oct 2007 15:57:29 GMT
View Forum Message <> Reply to Message

Just switch to the consolidated calls.

ipt_recent() has to initialize the private, so use
the __seq_open_private() helper.

Signed-off-by: Pavel Emelyanov <xemul@openvz.org>
Cc: Patrick McHardy <kaber@trash.net>

```
diff --git a/net/ipv4/netfilter/ipt_recent.c b/net/ipv4/netfilter/ipt_recent.c
index db2a798..11d39fb 100644
--- a/net/ipv4/netfilter/ipt_recent.c
+++ b/net/ipv4/netfilter/ipt_recent.c
@@ -381,25 +381,14 @@ static const struct seq_operations recen
 static int recent_seq_open(struct inode *inode, struct file *file)
 {
  struct proc_dir_entry *pde = PDE(inode);
- struct seq_file *seq;
  struct recent_iter_state *st;
- int ret;

- st = kzalloc(sizeof(*st), GFP_KERNEL);
+ st = __seq_open_private(file, &recent_seq_ops, sizeof(*st));
```

```
  if (st == NULL)
   return -ENOMEM;

- ret = seq_open(file, &recent_seq_ops);
- if (ret) {
-  kfree(st);
-  goto out;
- }
-
  st->table    = pde->data;
- seq          = file->private_data;
- seq->private = st;
-out:
- return ret;
+ return 0;
 }

 static ssize_t recent_proc_write(struct file *file, const char __user *input,
diff --git a/net/ipv4/netfilter/nf_conntrack_l3proto_ipv4_compat.c
b/net/ipv4/netfilter/nf_conntrack_l3proto_ipv4_compat.c
index a5ae2ea..741f3df 100644
--- a/net/ipv4/netfilter/nf_conntrack_l3proto_ipv4_compat.c
+++ b/net/ipv4/netfilter/nf_conntrack_l3proto_ipv4_compat.c
@@ -174,22 +174,8 @@ static const struct seq_operations ct_se

 static int ct_open(struct inode *inode, struct file *file)
 {
- struct seq_file *seq;
- struct ct_iter_state *st;
- int ret;
-
- st = kzalloc(sizeof(struct ct_iter_state), GFP_KERNEL);
- if (st == NULL)
-  return -ENOMEM;
- ret = seq_open(file, &ct_seq_ops);
- if (ret)
-  goto out_free;
- seq          = file->private_data;
- seq->private = st;
- return ret;
-out_free:
- kfree(st);
- return ret;
+ return seq_open_private(file, &ct_seq_ops,
+   sizeof(struct ct_iter_state));
 }

 static const struct file_operations ct_file_ops = {
```

```
@@ -291,22 +277,8 @@ static const struct seq_operations exp_s

 static int exp_open(struct inode *inode, struct file *file)
 {
- struct seq_file *seq;
- struct ct_expect_iter_state *st;
- int ret;
-
- st = kzalloc(sizeof(struct ct_expect_iter_state), GFP_KERNEL);
- if (!st)
-  return -ENOMEM;
- ret = seq_open(file, &exp_seq_ops);
- if (ret)
-  goto out_free;
- seq          = file->private_data;
- seq->private = st;
- return ret;
-out_free:
- kfree(st);
- return ret;
+ return seq_open_private(file, &exp_seq_ops,
+    sizeof(struct ct_expect_iter_state));
 }

 static const struct file_operations ip_exp_file_ops = {
diff --git a/net/netfilter/nf_conntrack_expect.c b/net/netfilter/nf_conntrack_expect.c
index 7a0ae36..175c8d1 100644
--- a/net/netfilter/nf_conntrack_expect.c
+++ b/net/netfilter/nf_conntrack_expect.c
@@ -472,22 +472,8 @@ static const struct seq_operations exp_s

 static int exp_open(struct inode *inode, struct file *file)
 {
- struct seq_file *seq;
- struct ct_expect_iter_state *st;
- int ret;
-
- st = kzalloc(sizeof(struct ct_expect_iter_state), GFP_KERNEL);
- if (!st)
-  return -ENOMEM;
- ret = seq_open(file, &exp_seq_ops);
- if (ret)
-  goto out_free;
- seq          = file->private_data;
- seq->private = st;
- return ret;
-out_free:
- kfree(st);
```

```
- return ret;
+ return seq_open_private(file, &exp_seq_ops,
+   sizeof(struct ct_expect_iter_state));
 }

 static const struct file_operations exp_file_ops = {
diff --git a/net/netfilter/nf_conntrack_standalone.c b/net/netfilter/nf_conntrack_standalone.c
index 2a19c5f..9efdd37 100644
--- a/net/netfilter/nf_conntrack_standalone.c
+++ b/net/netfilter/nf_conntrack_standalone.c
@@ -195,22 +195,8 @@ static const struct seq_operations ct_se

 static int ct_open(struct inode *inode, struct file *file)
 {
- struct seq_file *seq;
- struct ct_iter_state *st;
- int ret;
-
- st = kzalloc(sizeof(struct ct_iter_state), GFP_KERNEL);
- if (st == NULL)
-  return -ENOMEM;
- ret = seq_open(file, &ct_seq_ops);
- if (ret)
-  goto out_free;
- seq          = file->private_data;
- seq->private = st;
- return ret;
-out_free:
- kfree(st);
- return ret;
+ return seq_open_private(file, &ct_seq_ops,
+   sizeof(struct ct_iter_state));
 }

 static const struct file_operations ct_file_ops = {
diff --git a/net/netfilter/nfnetlink_log.c b/net/netfilter/nfnetlink_log.c
index 2135926..2c7bd2e 100644
--- a/net/netfilter/nfnetlink_log.c
+++ b/net/netfilter/nfnetlink_log.c
@@ -951,22 +951,8 @@ static const struct seq_operations nful_

 static int nful_open(struct inode *inode, struct file *file)
 {
- struct seq_file *seq;
- struct iter_state *is;
- int ret;
-
- is = kzalloc(sizeof(*is), GFP_KERNEL);
```

```
- if (!is)
-  return -ENOMEM;
- ret = seq_open(file, &nful_seq_ops);
- if (ret < 0)
-  goto out_free;
- seq = file->private_data;
- seq->private = is;
- return ret;
-out_free:
- kfree(is);
- return ret;
+ return seq_open_private(file, &nful_seq_ops,
+   sizeof(struct iter_state));
 }

 static const struct file_operations nful_file_ops = {
diff --git a/net/netfilter/nfnetlink_queue.c b/net/netfilter/nfnetlink_queue.c
index 48e095a..49f0480 100644
--- a/net/netfilter/nfnetlink_queue.c
+++ b/net/netfilter/nfnetlink_queue.c
@@ -1051,22 +1051,8 @@ static const struct seq_operations nfqnl

 static int nfqnl_open(struct inode *inode, struct file *file)
 {
- struct seq_file *seq;
- struct iter_state *is;
- int ret;
-
- is = kzalloc(sizeof(*is), GFP_KERNEL);
- if (!is)
-  return -ENOMEM;
- ret = seq_open(file, &nfqnl_seq_ops);
- if (ret < 0)
-  goto out_free;
- seq = file->private_data;
- seq->private = is;
- return ret;
-out_free:
- kfree(is);
- return ret;
+ return seq_open_private(file, &nfqnl_seq_ops,
+   sizeof(struct iter_state));
 }

 static const struct file_operations nfqnl_file_ops = {
```

Subject: [PATCH 4/6][NET-2.6.24] Make decnet code use the seq_open_private()
Posted by Pavel Emelianov on Tue, 09 Oct 2007 15:59:38 GMT
View Forum Message <> Reply to Message

Just switch to the consolidated code.

Signed-off-by: Pavel Emelyanov <xemul@openvz.org>
Cc: Patrick Caulfield <patrick@tykepenguin.com>

```
diff --git a/net/decnet/dn_neigh.c b/net/decnet/dn_neigh.c
index b66e3be..e851b14 100644
--- a/net/decnet/dn_neigh.c
+++ b/net/decnet/dn_neigh.c
@@ -580,24 +580,8 @@ static const struct seq_operations dn_ne

 static int dn_neigh_seq_open(struct inode *inode, struct file *file)
 {
- struct seq_file *seq;
- int rc = -ENOMEM;
- struct neigh_seq_state *s = kzalloc(sizeof(*s), GFP_KERNEL);
-
- if (!s)
-  goto out;
-
- rc = seq_open(file, &dn_neigh_seq_ops);
- if (rc)
-  goto out_kfree;
-
- seq          = file->private_data;
- seq->private = s;
-out:
- return rc;
-out_kfree:
- kfree(s);
- goto out;
+ return seq_open_private(file, &dn_neigh_seq_ops,
+    sizeof(struct neigh_seq_state));
 }

 static const struct file_operations dn_neigh_seq_fops = {
diff --git a/net/decnet/dn_route.c b/net/decnet/dn_route.c
index b7ebf99..97eee5e 100644
--- a/net/decnet/dn_route.c
+++ b/net/decnet/dn_route.c
@@ -1739,23 +1739,8 @@ static const struct seq_operations dn_rt

 static int dn_rt_cache_seq_open(struct inode *inode, struct file *file)
```

```
 {
- struct seq_file *seq;
- int rc = -ENOMEM;
- struct dn_rt_cache_iter_state *s;
-
- s = kzalloc(sizeof(*s), GFP_KERNEL);
- if (!s)
-  goto out;
- rc = seq_open(file, &dn_rt_cache_seq_ops);
- if (rc)
-  goto out_kfree;
- seq  = file->private_data;
- seq->private = s;
-out:
- return rc;
-out_kfree:
- kfree(s);
- goto out;
+ return seq_open_private(file, &dn_rt_cache_seq_ops,
+   sizeof(struct dn_rt_cache_iter_state));
 }

 static const struct file_operations dn_rt_cache_seq_fops = {
```

Subject: [PATCH 5/6][NET-2.6.24] Make the IRDA use the seq_open_private()
Posted by Pavel Emelianov on Tue, 09 Oct 2007 16:01:32 GMT
View Forum Message <> Reply to Message

Just switch to the consolidated code

Signed-off-by: Pavel Emelyanov <xemul@openvz.org>
Cc: Samuel Ortiz <samuel@sortiz.org>

```
diff --git a/net/irda/irlap.c b/net/irda/irlap.c
index 3d76aaf..f3236ac 100644
--- a/net/irda/irlap.c
+++ b/net/irda/irlap.c
@@ -1219,29 +1219,11 @@ static const struct seq_operations irlap

 static int irlap_seq_open(struct inode *inode, struct file *file)
 {
- struct seq_file *seq;
- int rc = -ENOMEM;
- struct irlap_iter_state *s = kzalloc(sizeof(*s), GFP_KERNEL);
+ if (irlap == NULL)
```

```
+  return -EINVAL;

- if (!s)
-  goto out;
-
- if (irlap == NULL) {
-  rc = -EINVAL;
-  goto out_kfree;
- }
-
- rc = seq_open(file, &irlap_seq_ops);
- if (rc)
-  goto out_kfree;
-
- seq      = file->private_data;
- seq->private = s;
-out:
- return rc;
-out_kfree:
- kfree(s);
- goto out;
+ return seq_open_private(file, &irlap_seq_ops,
+   sizeof(struct irlap_iter_state));
 }

 const struct file_operations irlap_seq_fops = {
diff --git a/net/irda/irlmp.c b/net/irda/irlmp.c
index 7efa930..7db92ce 100644
--- a/net/irda/irlmp.c
+++ b/net/irda/irlmp.c
@@ -2003,27 +2003,10 @@ static const struct seq_operations irlmp

 static int irlmp_seq_open(struct inode *inode, struct file *file)
 {
- struct seq_file *seq;
- int rc = -ENOMEM;
- struct irlmp_iter_state *s;
-
  IRDA_ASSERT(irlmp != NULL, return -EINVAL;);

- s = kmalloc(sizeof(*s), GFP_KERNEL);
- if (!s)
-  goto out;
-
- rc = seq_open(file, &irlmp_seq_ops);
- if (rc)
-  goto out_kfree;
-
```

```
- seq      = file->private_data;
- seq->private = s;
-out:
- return rc;
-out_kfree:
- kfree(s);
- goto out;
+ return seq_open_private(file, &irlmp_seq_ops,
+   sizeof(struct irlmp_iter_state));
 }

 const struct file_operations irlmp_seq_fops = {
diff --git a/net/irda/irttp.c b/net/irda/irttp.c
index 3d7ab03..1311976 100644
--- a/net/irda/irttp.c
+++ b/net/irda/irttp.c
@@ -1884,25 +1884,8 @@ static const struct seq_operations irttp

 static int irttp_seq_open(struct inode *inode, struct file *file)
 {
- struct seq_file *seq;
- int rc = -ENOMEM;
- struct irttp_iter_state *s;
-
- s = kzalloc(sizeof(*s), GFP_KERNEL);
- if (!s)
-  goto out;
-
- rc = seq_open(file, &irttp_seq_ops);
- if (rc)
-  goto out_kfree;
-
- seq      = file->private_data;
- seq->private = s;
-out:
- return rc;
-out_kfree:
- kfree(s);
- goto out;
+ return seq_open_private(file, &irttp_seq_ops,
+   sizeof(struct irttp_iter_state));
 }

 const struct file_operations irttp_seq_fops = {
```

Subject: [PATCH 6/6][NET-2.6.24] Make the sunrpc use the seq_open_private()

Posted by Pavel Emelianov on Tue, 09 Oct 2007 16:04:23 GMT
View Forum Message <> Reply to Message

Just switch to the consolidated code.

Signed-off-by: Pavel Emelyanov <xemul@openvz.org>
Cc: Neil Brown <neilb@cse.unsw.edu.au>

```
diff --git a/net/sunrpc/cache.c b/net/sunrpc/cache.c
index ebe344f..8e05557 100644
--- a/net/sunrpc/cache.c
+++ b/net/sunrpc/cache.c
@@ -1218,23 +1218,15 @@ static const struct seq_operations cache

 static int content_open(struct inode *inode, struct file *file)
 {
- int res;
  struct handle *han;
  struct cache_detail *cd = PDE(inode)->data;

- han = kmalloc(sizeof(*han), GFP_KERNEL);
+ han = __seq_open_private(file, &cache_content_op, sizeof(*han));
  if (han == NULL)
   return -ENOMEM;

  han->cd = cd;
-
- res = seq_open(file, &cache_content_op);
- if (res)
-  kfree(han);
- else
-  ((struct seq_file *)file->private_data)->private = han;
-
- return res;
+ return 0;
 }

 static const struct file_operations content_file_operations = {
```

Subject: Re: [PATCH 1/6][NET-2.6.24] Introduce the seq_open_private()
Posted by davem on Wed, 10 Oct 2007 09:32:52 GMT
View Forum Message <> Reply to Message

From: Pavel Emelyanov <xemul@openvz.org>
Date: Tue, 09 Oct 2007 19:52:58 +0400

> This function allocates the zeroed chunk of memory and
> call seq_open(). The __seq_open_private() helper returns
> the allocated memory to make it possible for the caller
> to initialize it.
>
> Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

Applied, nice cleanup Pavel.

Subject: Re: [PATCH 2/6][NET-2.6.24] Make core networking code use seq_open_private
Posted by davem on Wed, 10 Oct 2007 09:32:59 GMT
View Forum Message <> Reply to Message

From: Pavel Emelyanov <xemul@openvz.org>
Date: Tue, 09 Oct 2007 19:55:28 +0400

> This concerns the ipv4 and ipv6 code mostly, but also the netlink
> and unix sockets.
>
> The netlink code is an example of how to use the __seq_open_private()
> call - it saves the net namespace on this private.
>
> Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

Applied.

Subject: Re: [PATCH 3/6][NET-2.6.24] Make netfilter code use the seq_open_private
Posted by davem on Wed, 10 Oct 2007 09:33:15 GMT
View Forum Message <> Reply to Message

From: Pavel Emelyanov <xemul@openvz.org>
Date: Tue, 09 Oct 2007 19:57:29 +0400

> Just switch to the consolidated calls.
>
> ipt_recent() has to initialize the private, so use
> the __seq_open_private() helper.
>
> Signed-off-by: Pavel Emelyanov <xemul@openvz.org>
> Cc: Patrick McHardy <kaber@trash.net>

Applied.

Subject: Re: [PATCH 4/6][NET-2.6.24] Make decnet code use the seq_open_private()
Posted by davem on Wed, 10 Oct 2007 09:33:23 GMT
View Forum Message <> Reply to Message

From: Pavel Emelyanov <xemul@openvz.org>
Date: Tue, 09 Oct 2007 19:59:38 +0400

> Just switch to the consolidated code.
>
> Signed-off-by: Pavel Emelyanov <xemul@openvz.org>
> Cc: Patrick Caulfield <patrick@tykepenguin.com>

Applied.

Subject: Re: [PATCH 5/6][NET-2.6.24] Make the IRDA use the seq_open_private()
Posted by davem on Wed, 10 Oct 2007 09:33:29 GMT
View Forum Message <> Reply to Message

From: Pavel Emelyanov <xemul@openvz.org>
Date: Tue, 09 Oct 2007 20:01:32 +0400

> Just switch to the consolidated code
>
> Signed-off-by: Pavel Emelyanov <xemul@openvz.org>
> Cc: Samuel Ortiz <samuel@sortiz.org>

Applied.

Subject: Re: [PATCH 6/6][NET-2.6.24] Make the sunrpc use the seq_open_private()
Posted by davem on Wed, 10 Oct 2007 09:33:35 GMT
View Forum Message <> Reply to Message

From: Pavel Emelyanov <xemul@openvz.org>
Date: Tue, 09 Oct 2007 20:04:23 +0400

> Just switch to the consolidated code.
>
> Signed-off-by: Pavel Emelyanov <xemul@openvz.org>
> Cc: Neil Brown <neilb@cse.unsw.edu.au>

Applied.
