

new/usr/src/uts/common/vm/seg_spt.c

1

```
*****
82318 Fri May 8 18:05:05 2015
new/usr/src/uts/common/vm/seg_spt.c
segspt_ops can be static
There is nothing that needs access to this structure outside of the spt
segment driver itself.
*****
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License (the "License").
6  * You may not use this file except in compliance with the License.
7  *
8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  * or http://www.opensolaris.org/os/licensing.
10 * See the License for the specific language governing permissions
11 * and limitations under the License.
12 *
13 * When distributing Covered Code, include this CDDL HEADER in each
14 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 * If applicable, add the following below this CDDL HEADER, with the
16 * fields enclosed by brackets "[]" replaced with your own identifying
17 * information: Portions Copyright [yyyy] [name of copyright owner]
18 *
19 * CDDL HEADER END
20 */
21 /*
22 * Copyright (c) 1993, 2010, Oracle and/or its affiliates. All rights reserved.
23 */

25 #include <sys/param.h>
26 #include <sys/user.h>
27 #include <sys/mman.h>
28 #include <sys/kmem.h>
29 #include <sys/sysmacros.h>
30 #include <sys/cmn_err.h>
31 #include <sys/system.h>
32 #include <sys/tuneable.h>
33 #include <vm/hat.h>
34 #include <vm/seg.h>
35 #include <vm/as.h>
36 #include <vm/anon.h>
37 #include <vm/page.h>
38 #include <sys/buf.h>
39 #include <sys/swap.h>
40 #include <sys/atomic.h>
41 #include <vm/seg_spt.h>
42 #include <sys/debug.h>
43 #include <sys/vtrace.h>
44 #include <sys/shm.h>
45 #include <sys/shm_impl.h>
46 #include <sys/lgrp.h>
47 #include <sys/vmsystem.h>
48 #include <sys/policy.h>
49 #include <sys/project.h>
50 #include <sys/tnf_probe.h>
51 #include <sys/zone.h>

53 #define SEGSPADDR      (caddr_t)0x0

55 /*
56  * # pages used for spt
57  */
58 size_t  spt_used;
```

new/usr/src/uts/common/vm/seg_spt.c

2

```
60 /*
61  * segspt_minfree is the memory left for system after ISM
62  * locked its pages; it is set up to 5% of availrmem in
63  * sptcreate when ISM is created.  ISM should not use more
64  * than ~90% of availrmem; if it does, then the performance
65  * of the system may decrease.  Machines with large memories may
66  * be able to use up more memory for ISM so we set the default
67  * segspt_minfree to 5% (which gives ISM max 95% of availrmem.
68  * If somebody wants even more memory for ISM (risking hanging
69  * the system) they can patch the segspt_minfree to smaller number.
70  */
71 pgcnt_t segspt_minfree = 0;

73 static int segspt_create(struct seg *seg, caddr_t argsp);
74 static int segspt_unmap(struct seg *seg, caddr_t raddr, size_t ssize);
75 static void segspt_free(struct seg *seg);
76 static void segspt_free_pages(struct seg *seg, caddr_t addr, size_t len);
77 static lgrp_mem_policy_info_t *segspt_getpolicy(struct seg *seg, caddr_t addr);

79 static const struct seg_ops segspt_ops = {
79 const struct seg_ops segspt_ops = {
80     .unmap      = segspt_unmap,
81     .free       = segspt_free,
82     .getpolicy  = segspt_getpolicy,
83 };
unchanged_portion_omitted
```