

```

*****
1966 Wed Nov 5 12:56:19 2014
new/usr/src/uts/intel/asm/bitmap.h
patch bitmap
*****
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, Version 1.0 only
6  * (the "License"). You may not use this file except in compliance
7  * with the License.
8  *
9  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 * or http://www.opensolaris.org/os/licensing.
11 * See the License for the specific language governing permissions
12 * and limitations under the License.
13 *
14 * When distributing Covered Code, include this CDDL HEADER in each
15 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 * If applicable, add the following below this CDDL HEADER, with the
17 * fields enclosed by brackets "[]" replaced with your own identifying
18 * information: Portions Copyright [yyyy] [name of copyright owner]
19 *
20 * CDDL HEADER END
21 */
22 /*
23 * Copyright 2005 Sun Microsystems, Inc. All rights reserved.
24 * Use is subject to license terms.
25 */

27 #ifndef _ASM_BITMAP_H
28 #define _ASM_BITMAP_H

30 #include <sys/ccompil.h>
31 #include <sys/types.h>

33 #ifdef __cplusplus
34 extern "C" {
35 #endif

37 #if !defined(__lint) && defined(__GNUC__)

39 #if defined(__amd64)
40 #define __SUF "q"
41 #elif defined(__i386)
42 #define __SUF "l"
43 #else
44 #error "port me"
45 #endif

47 #endif /* !codereview */
48 extern __GNU_INLINE int
49 highbit(ulong_t i)
50 {
51     long value = -1l;
39     long __value = -1l;

41 #if defined(__amd64)
53     __asm__(
54         "bsr" __SUF " %1,%0"
55         : "+r" (value)
56         : "bsrq" __SUF " %1,%0"
57         : "+r" (__value)
58         : "r" (i)
59         : "cc");

```

```

59     return ((int)(value + 1));
47 #elif defined(__i386)
48     __asm__(
49         "bsrl" __SUF " %1,%0"
50         : "+r" (__value)
51         : "r" (i)
52         : "cc");
53 #else
54 #error "port me"
55 #endif
56     return ((int)(__value + 1));
60 }

62 extern __GNU_INLINE int
63 lowbit(ulong_t i)
64 {
65     long value = -1l;
62     long __value = -1l;

64 #if defined(__amd64)
66     __asm__(
67         "bsf" __SUF " %1,%0"
68         : "+r" (value)
69         : "bsfq" __SUF " %1,%0"
70         : "+r" (__value)
71         : "r" (i)
72         : "cc");

73     return ((int)(value + 1));
70 #elif defined(__i386)
71     __asm__(
72         "bsfl" __SUF " %1,%0"
73         : "+r" (__value)
74         : "r" (i)
75         : "cc");
76 #else
77 #error "port me"
78 #endif
79     return ((int)(__value + 1));
74 }

76 extern __GNU_INLINE uint_t
77 atomic_btr32(uint32_t *memory, uint_t bitnum)
78 {
79     uint8_t value;
85     uint8_t __value;

87 #if defined(__amd64)
81     __asm__ __volatile__(
82         "lock;"
83         "btrl %2,%0;"
90         "btrl %2, (%0);"
84         "setc %1"
85         : "+m" (*memory), "=r" (value)
92         : "+r" (memory), "+r" (__value)
86         : "ir" (bitnum)
87         : "cc");

89     return ((uint_t)value);
95 #elif defined(__i386)
96     __asm__ __volatile__(
97         "lock;"
98         "btrl %2, (%0);"
99         "setc %1"
100        : "+r" (memory), "=r" (__value)

```

```
101         : "ir" (bitnum)
102         : "cc");
103 #else
104 #error "port me"
105 #endif
106         return ((uint_t)_value);
107     }

```

```
92 #undef __SUF
93 #endif /* !codereview */

95 #endif /* !__lint && __GNUC__ */

97 #ifdef __cplusplus
98 }
99 #endif

101 #endif /* _ASM_BITMAP_H */

```