9.11

虚拟地址：0x027c

A．虚拟地址格式

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |

B.地址翻译

|  |  |
| --- | --- |
| 参数 | 值 |
| VPN | 0x09 |
| TLB索引 | 0x01 |
| TLB标记 | 0x02 |
| TLB命中？（是/否） | 否 |
| 缺页？（是/否） | 否 |
| PPN | 0x17 |

C.物理地址格式

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |

D．物理地址引用

|  |  |
| --- | --- |
| 参数 | 值 |
| 字节偏移 | 0x00 |
| 缓存索引 | 0x0f |
| 缓存标记 | 0x17 |
| 缓存命中？（是/否） | 否 |
| 返回的缓存字节 | - |

9.13

虚拟地址：0x0040

A．虚拟地址格式

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

B.地址翻译

|  |  |
| --- | --- |
| 参数 | 值 |
| VPN | 0x01 |
| TLB索引 | 0x01 |
| TLB标记 | 0x00 |
| TLB命中？（是/否） | 否 |
| 缺页？（是/否） | 是 |
| PPN | - |

C.物理地址格式

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|  |  |  |  |  |  |  |  |  |  |  |  |

D．物理地址引用

|  |  |
| --- | --- |
| 参数 | 值 |
| 字节偏移 |  |
| 缓存索引 |  |
| 缓存标记 |  |
| 缓存命中？（是/否） |  |
| 返回的缓存字节 |  |

9.15

|  |  |  |
| --- | --- | --- |
| 请求 | 块大小（十进制字节） | 块头部（十六进制） |
| malloc(3) | 8 | 0x00000009 |
| malloc(11) | 16 | 0x00000011 |
| malloc(20) | 24 | 0x00000019 |
| malloc(21) | 32 | 0x00000021 |

9.17

绝大多数函数的实现同lab8中所给代码，但由于采用下一次适配搜索，个别函数（mm\_init、find\_fit、coalesce）稍作修改（其实就是修改了条件编译的各个部分）如下：

/\*

\* mm\_init - Initialize the memory manager

\*/

/\* $begin mminit \*/

int mm\_init(void)

{

/\* create the initial empty heap \*/

if ((heap\_listp = mem\_sbrk(4\*WSIZE)) == NULL)

return -1;

PUT(heap\_listp, 0); /\* alignment padding \*/

PUT(heap\_listp+WSIZE, PACK(OVERHEAD, 1)); /\* prologue header \*/

PUT(heap\_listp+DSIZE, PACK(OVERHEAD, 1)); /\* prologue footer \*/

PUT(heap\_listp+WSIZE+DSIZE, PACK(0, 1)); /\* epilogue header \*/

heap\_listp += DSIZE;

rover = heap\_listp;

/\* Extend the empty heap with a free block of CHUNKSIZE bytes \*/

if (extend\_heap(CHUNKSIZE/WSIZE) == NULL)

return -1;

return 0;

}

/\* $end mminit \*/

/\*

\* find\_fit - Find a fit for a block with asize bytes

\*/

static void \*find\_fit(size\_t asize)

{

/\* next fit search \*/

char \*oldrover = rover;

/\* search from the rover to the end of list \*/

for ( ; GET\_SIZE(HDRP(rover)) > 0; rover = NEXT\_BLKP(rover))

if (!GET\_ALLOC(HDRP(rover)) && (asize <= GET\_SIZE(HDRP(rover))))

return rover;

/\* search from start of list to old rover \*/

for (rover = heap\_listp; rover < oldrover; rover = NEXT\_BLKP(rover))

if (!GET\_ALLOC(HDRP(rover)) && (asize <= GET\_SIZE(HDRP(rover))))

return rover;

return NULL; /\* no fit found \*/

}

/\*

\* coalesce - boundary tag coalescing. Return ptr to coalesced block

\*/

static void \*coalesce(void \*bp)

{

size\_t prev\_alloc = GET\_ALLOC(FTRP(PREV\_BLKP(bp)));

size\_t next\_alloc = GET\_ALLOC(HDRP(NEXT\_BLKP(bp)));

size\_t size = GET\_SIZE(HDRP(bp));

if(prev\_alloc&&next\_alloc) /\*Case 1\*/

{

return bp;

}

else if(prev\_alloc && !next\_alloc) /\*Case 2\*/

{

size += GET\_SIZE(HDRP(NEXT\_BLKP(bp)));

PUT(HDRP(bp),PACK(size,0));

PUT(FTRP(bp),PACK(size,0));

}

else if(!prev\_alloc && next\_alloc) /\*Case 3\*/

{

size += GET\_SIZE(HDRP(PREV\_BLKP(bp)));

PUT(FTRP(bp),PACK(size,0));

PUT(HDRP(PREV\_BLKP(bp)),PACK(size,0));

bp = PREV\_BLKP(bp);

}

else /\*Case 4\*/

{

size += GET\_SIZE(HDRP(PREV\_BLKP(bp))) + GET\_SIZE(FTRP(NEXT\_BLKP(bp)));

PUT(HDRP(PREV\_BLKP(bp)),PACK(size,0));

PUT(FTRP(NEXT\_BLKP(bp)),PACK(size,0));

bp = PREV\_BLKP(bp);

}

if ((rover > (char \*)bp) && (rover < NEXT\_BLKP(bp))) /\*make rover not point to the free block that just coalesced\*/

rover = bp;

return bp;

}

9.19

1) a)

2) d)

3) b)