Lab1文档

“任务与思考题”部分仅直截了当记录实验结果并回答问题，具体实验细节在“实验过程”部分

任务与思考题

1.内存分配大小分析

请求分配大小	实际分配大小	分析与说明
96 字节	128字节	96+32（meta大小）<819字节，使用salloc进行分配，向上取整分配1*128字节的小页面
128 字节	256字节	128+32（meta大小）<819字节，使用salloc进行分配，向上取整分配2*128字节的小页面
256 字节	384字节	256+32（meta大小）<819字节，使用salloc进行分配，向上取整分配3*128字节的小页面
4064 字节	4096字节	4064+32（meta大小）>819字节，使用palloc进行分配，向上取整分配1*4096字节的页面
4096 字节	8192字节	4096+32（meta大小）>819字节，使用palloc进行分配，向上取整分配2*4096字节的页面

2.核心问题

（1）最小分配单元: Unikraft 两种内存分配策略的最小单元是多少？它是如何定义的?

Unikraft采用salloc和palloc两种内存分配策略，其中：

- salloc分配的最小单元为128字节，在__S_PAGE_SIZE常量中定义，salloc时根据请求大小向上取整分配若干128字节大小的页面
- palloc分配的最小单元为4096字节，在__PAGE_SIZE常量中定义，palloc时根据请求大小向上取整分配若干4096字节大小的页面

（2）分配器选择：uk_malloc() 函数在何种条件下会选择 palloc，又在何种条件下会选择salloc?

在处理大小为size的分配请求时，首先计算size+METADATA_IFPAGES_SIZE_POW2（meta数据的大小）是否<__PAGE_SIZE/5（819），如果分配请求真实需要分配的大小<819个字节则使用salloc进行分配，否则使用palloc进行分配

（3）大内存分配问题: 当前 palloc 在处理大内存（例如，一次性分配多个页面）的分配与回收时，存在一个已知的设计问题。请定位该问题，并尝试在 GDB 中通过 set 命令修改相关变量，模拟正确的 free 过程，并截图记录结果。

问题出现在small参数没有被正确进行初始化（是一个很奇怪的指针值），导致在根据small参数判断“究竟该使用大页面的释放方式还是小页面的释放方式”时，模型总是会选择按照小页面的方式进行释放，导致使用palloc申请的内存区域无法正确释放。共做出两处改动：

1.在alloc.c设置is_small参数为0后，成功调用大页面的PAGE_ALIGN_DOWN函数而不是小页面的PAGE_ALIGN_DOWN_S

```
    at /workspaces/OS-2026/app-helloworld/workdir/unikraft/lib/ukalloc/alloc.c:116
116       UK_ASSERT((__uptr)ptr >= __PAGE_SIZE + METADATA_IFPAGES_SIZE_POW2);
(gdb) s
119       int is_small = (int)small;
(gdb) s
120       if (is_small) {
(gdb) set variable is_small = 0
(gdb) p is_small
$1 = 0
(gdb) s
130         metadata = PAGE_ALIGN_DOWN((__uptr)ptr);
(gdb) s
131         if (metadata == (__uptr)ptr) {
```

2.在判断sfree和pfree之前手动设置small参数为0，成功调用uk_pfree函数而不是uk_sfree函数进行内存释放

```
(gdb) p *small
Attempt to dereference a generic pointer.
(gdb) set variable small = 0
(gdb) p small
$4 = (const void *) 0x0
(gdb) s
224         uk_pfree(a, metadata->base, metadata->num_pages);
(gdb) s
uk_pfree (a=0x40010000, ptr=0x40192000, num_pages=1)
    at /workspaces/OS-2026/app-helloworld/workdir/unikraft/lib/ukalloc/include/uk/alloc.h:249
249       uk_do_pfree(a, ptr, num_pages);
(gdb) s
uk_do_pfree (a=0x40010000, ptr=0x40192000, num_pages=1)
    at /workspaces/OS-2026/app-helloworld/workdir/unikraft/lib/ukalloc/include/uk/alloc.h:238
238       UK_ASSERT(a);
```

关于手动设置is_small参数之后，free函数的表现差异（内存变化结果）呈现在实验过程章节

实验过程

1.内存分配大小分析

（1）在main.c malloc和free的位置设置断点，对于malloc其调用链可追溯为：malloc -> uk_malloc -> uk_do_malloc -> uk_malloc_ifpages；对于uk_malloc_ifpages函数，对其注释分析如下：

代码块

```
void *uk_malloc_ifpages(struct uk_alloc *a, __sz size) {
  __uptr intptr;
  unsigned long num_pages;
  struct metadata_ifpages *metadata;
#ifdef CONFIG_HAVE_MEMTAG
  size = MEMTAG_ALIGN(size);
#endif
  /*由于需要空间存储分配内存块的元数据信息，因而这里需要添加一个固定大小的常量METADATA_IFPAGES_SIZE_POW2（值为32）*/
  __sz realsize = METADATA_IFPAGES_SIZE_POW2 + size;
  UK_ASSERT(a);
  if (!size || realsize < size) return __NULL;
  /*这里使用IS_SMALL函数判断真实待分配区域的大小是否<__PAGE_SIZE/5，其中__PAGE_SIZE为4096，所以临界点为819字节，大于等于这个临界点使用palloc，否则使用salloc*/
  if (IS_SMALL(realsize)) {
    /*判断需要几个128字节的小页面，并调用uk_salloc函数进行分配*/
    num_pages = size_to_s_num_pages(realsize);
    intptr = (__uptr)uk_salloc(a, num_pages);
    uk_pr_err("alloc size => %llu, num_pages => %llu, intpter => %p\n",
              realsize, num_pages, intptr);
  }
  /*判断需要几个4096字节的大页面，并调用uk_palloc函数进行分配*/
  else {
    num_pages = size_to_num_pages(realsize);
    intptr = (__uptr)uk_palloc(a, num_pages);
  }
  if (!intptr) return __NULL;
  /*这里在最开始32个字节内的区域设置元信息，包括大小、起始地址和分配的页数*/
  metadata = (struct metadata_ifpages *)intptr;
  metadata->size = size;
  metadata->num_pages = num_pages;
  metadata->base = (void *)intptr;

#ifdef CONFIG_HAVE_MEMTAG
  return ukarch_memtag_region((void *)(intptr + METADATA_IFPAGES_SIZE_POW2),
                              size);
#else
  /*由于分配区域前32个字节是meta数据，因而程序实际可用的是32字节以后的位置*/
  return (void *)(intptr + METADATA_IFPAGES_SIZE_POW2);
#endif
```

```
39   }
```

（2）在对调用salloc/palloc进行内存分配后，我们可以使用gdb查看各内存块metadata的情况：

对于salloc的情况（96/128/256字节）：

```
(gdb) p *metadata
$6 = {size = 96, num_pages = 1, base = 0x44101a00}
```

```
(gdb) p *metadata
$3 = {size = 128, num_pages = 2, base = 0x44101980}
```

```
(gdb) p *metadata
$2 = {size = 256, num_pages = 3, base = 0x44101900}
```

对于palloc的情况（4064/4096字节）：

```
(gdb) p *metadata
$2 = {size = 4064, num_pages = 1, base = 0x40192000}
```

```
199       return (void *)(intptr + METADATA_IFPAGES_SIZE_POW2);
(gdb) p *metadata
$2 = {size = 4096, num_pages = 2, base = 0x40192000}
```

（3）同样地，对在free位置的断点进行深入，直到uk_free_ifpages函数，对该函数进行注释分析如下：

代码块

```
1   void uk_free_ifpages(struct uk_alloc *a, void *ptr, const void *small) {
2     struct metadata_ifpages *metadata;
3   #ifdef CONFIG_HAVE_MEMTAG
4     __sz size;
5   #endif
6     UK_ASSERT(a);
7     if (!ptr) return;
8   #ifdef CONFIG_HAVE_MEMTAG
9     metadata = uk_get_metadata((void *)((__u64)ptr & ~MTE_TAG_MASK));
10    size = metadata->size;
11  #else
12    /*获得该内存块的元数据信息（页数、起始地址等）*/
13    metadata = uk_get_metadata(ptr, small);
14  #endif
15    UK_ASSERT(metadata->base != __NULL);
16    UK_ASSERT(metadata->num_pages != 0);
17    /*如果是128字节的小页面，调用uk_sfree释放metadata中记录的num_pages个页面*/
18    if (small) {
19      uk_sfree(a, metadata->base, metadata->num_pages);
20    }
21    /*否则调用uk_pfree释放记录的num_pages个4096字节大小的正常页面*/
```

```
    else {
      uk_pfree(a, metadata->base, metadata->num_pages);
    }

#ifdef CONFIG_HAVE_MEMTAG
    ukarch_memtag_region(ptr, size);
#endif
}
```

2.free算法流程模拟

理论上，在调用malloc(4096)前和free之后，内存空闲情况应保持一致

在未改动代码情况下，由于small参数未正确初始化，导致内存申请释放前后系统内存状态不一致（左边为申请前，右边为释放后）

代码块

```
[    4.239707] ERR: [libukallocbbuddy] <bbuddy.c @ 305> Dumping current state of the free list:
[    4.240418] ERR: [libukallocbbuddy] <bbuddy.c @ 320> Free list 0 is empty.
[    4.240992] ERR: [libukallocbbuddy] <bbuddy.c @ 322> Free list 1 (Order 1):
[    4.241536] ERR: [libukallocbbuddy] <bbuddy.c @ 326>   Entry at address: 0x40192000, end: 0x40194000, level: 1
[    4.242440] ERR: [libukallocbbuddy] <bbuddy.c @ 322> Free list 2 (Order 2):
[    4.242999] ERR: [libukallocbbuddy] <bbuddy.c @ 326>   Entry at address: 0x44104000, end: 0x44108000, level: 2
[    4.243844] ERR: [libukallocbbuddy] <bbuddy.c @ 326>   Entry at address:
```

代码块

```
[    4.292088] ERR: [libukallocbbuddy] <bbuddy.c @ 345> Dumping current state of the sfree list:
[    4.292664] ERR: [libukallocbbuddy] <bbuddy.c @ 350> SFree list 0 (Order 0):
[    4.293150] ERR: [libukallocbbuddy] <bbuddy.c @ 354>   Entry at address: 0x44101a00, end: 0x44101a80, level: 0
[    4.293851] ERR: [libukallocbbuddy] <bbuddy.c @ 354>   Entry at address: 0x44101980, end: 0x44101a00, level: 0
[    4.294493] ERR: [libukallocbbuddy] <bbuddy.c @ 354>   Entry at address: 0x44101900, end: 0x44101980, level: 0
[    4.294906] ERR: [libukallocbbuddy] <bbuddy.c @ 354>   Entry at address: 0x44101880, end: 0x44101900, level: 0
```

```
0x40194000, end: 0x40198000, level: 2
[    4.244281] ERR: [libukallocbbuddy] <bbuddy.c @ 326>   Entry at address: 0x40014000, end: 0x40018000, level: 2
[    4.244733] ERR: [libukallocbbuddy] <bbuddy.c @ 322> Free list 3 (Order 3):
[    4.245043] ERR: [libukallocbbuddy] <bbuddy.c @ 326>   Entry at address: 0x44108000, end: 0x44110000, level: 3
[    4.245496] ERR: [libukallocbbuddy] <bbuddy.c @ 326>   Entry at address: 0x40198000, end: 0x401a0000, level: 3
[    4.245893] ERR: [libukallocbbuddy] <bbuddy.c @ 326>   Entry at address: 0x40018000, end: 0x40020000, level: 3
[    4.246510] ERR: [libukallocbbuddy] <bbuddy.c @ 322> Free list 4 (Order 4):
[    4.247293] ERR: [libukallocbbuddy] <bbuddy.c @ 326>   Entry at address: 0x44110000, end: 0x44120000, level: 4
[    4.248391] ERR: [libukallocbbuddy] <bbuddy.c @ 320> Free list 5 is empty.
[    4.249143] ERR: [libukallocbbuddy] <bbuddy.c @ 322> Free list 6 (Order 6):
[    4.249873] ERR: [libukallocbbuddy] <bbuddy.c @ 326>   Entry at address: 0x44140000, end: 0x44180000, level: 6
[    4.250499] ERR: [libukallocbbuddy] <bbuddy.c @ 326>   Entry at address:
```

```
[    4.295382] ERR: [libukallocbbuddy] <bbuddy.c @ 354>   Entry at address: 0x44101800, end: 0x44101880, level: 0
[    4.295789] ERR: [libukallocbbuddy] <bbuddy.c @ 354>   Entry at address: 0x44101780, end: 0x44101800, level: 0
[    4.296604] ERR: [libukallocbbuddy] <bbuddy.c @ 354>   Entry at address: 0x44101700, end: 0x44101780, level: 0
[    4.297248] ERR: [libukallocbbuddy] <bbuddy.c @ 354>   Entry at address: 0x44101680, end: 0x44101700, level: 0
[    4.298089] ERR: [libukallocbbuddy] <bbuddy.c @ 354>   Entry at address: 0x44101600, end: 0x44101680, level: 0
[    4.298583] ERR: [libukallocbbuddy] <bbuddy.c @ 354>   Entry at address: 0x44101580, end: 0x44101600, level: 0
```

```
    0x401c0000, end: 0x40200000,
    level: 6
19  [    4.251346] ERR:
    [libukallocbbuddy] <bbuddy.c @
    326>   Entry at address:
    0x40040000, end: 0x40080000,
    level: 6
20  [    4.253035] ERR:
    [libukallocbbuddy] <bbuddy.c @
    322> Free list 7 (Order 7):
21  [    4.253628] ERR:
    [libukallocbbuddy] <bbuddy.c @
    326>   Entry at address:
    0x44180000, end: 0x44200000,
    level: 7
22  [    4.254288] ERR:
    [libukallocbbuddy] <bbuddy.c @
    326>   Entry at address:
    0x40080000, end: 0x40100000,
    level: 7
23  [    4.256023] ERR:
    [libukallocbbuddy] <bbuddy.c @
    320> Free list 8 is empty.
24  [    4.256357] ERR:
    [libukallocbbuddy] <bbuddy.c @
    322> Free list 9 (Order 9):
25  [    4.256685] ERR:
    [libukallocbbuddy] <bbuddy.c @
    326>   Entry at address:
    0x44200000, end: 0x44400000,
    level: 9
26  [    4.257147] ERR:
    [libukallocbbuddy] <bbuddy.c @
    326>   Entry at address:
    0x40200000, end: 0x40400000,
    level: 9
```

在使用set命令模拟正确行为（修正small参数之后），内存申请释放前后系统内存状态一致（左边为申请前，右边为申请后）

代码块

```
1   [    4.276722] ERR:
    [libukallocbbuddy] <bbuddy.c @
    305> Dumping current state of
    the free list:
```

代码块

```
1   [    4.344764] ERR:
    [libukallocbbuddy] <bbuddy.c @
    305> Dumping current state of
    the free list:
```

```
[    4.277390] ERR: [libukallocbbuddy] <bbuddy.c @ 320> Free list 0 is empty.
[    4.277991] ERR: [libukallocbbuddy] <bbuddy.c @ 322> Free list 1 (Order 1):
[    4.278576] ERR: [libukallocbbuddy] <bbuddy.c @ 326>   Entry at address: 0x40192000, end: 0x40194000, level: 1
[    4.279332] ERR: [libukallocbbuddy] <bbuddy.c @ 322> Free list 2 (Order 2):
[    4.279942] ERR: [libukallocbbuddy] <bbuddy.c @ 326>   Entry at address: 0x44104000, end: 0x44108000, level: 2
[    4.280778] ERR: [libukallocbbuddy] <bbuddy.c @ 326>   Entry at address: 0x40194000, end: 0x40198000, level: 2
[    4.281586] ERR: [libukallocbbuddy] <bbuddy.c @ 326>   Entry at address: 0x40014000, end: 0x40018000, level: 2
[    4.282453] ERR: [libukallocbbuddy] <bbuddy.c @ 322> Free list 3 (Order 3):
[    4.283121] ERR: [libukallocbbuddy] <bbuddy.c @ 326>   Entry at address: 0x44108000, end: 0x44110000, level: 3
[    4.283618] ERR: [libukallocbbuddy] <bbuddy.c @ 326>   Entry at address: 0x40198000, end: 0x401a0000, level: 3
[    4.284389] ERR: [libukallocbbuddy] <bbuddy.c @ 326>   Entry at address: 0x40018000, end: 0x40020000, level: 3
```

```
[    4.345266] ERR: [libukallocbbuddy] <bbuddy.c @ 320> Free list 0 is empty.
[    4.345655] ERR: [libukallocbbuddy] <bbuddy.c @ 322> Free list 1 (Order 1):
[    4.346250] ERR: [libukallocbbuddy] <bbuddy.c @ 326>   Entry at address: 0x40192000, end: 0x40194000, level: 1
[    4.346900] ERR: [libukallocbbuddy] <bbuddy.c @ 322> Free list 2 (Order 2):
[    4.347215] ERR: [libukallocbbuddy] <bbuddy.c @ 326>   Entry at address: 0x44104000, end: 0x44108000, level: 2
[    4.348053] ERR: [libukallocbbuddy] <bbuddy.c @ 326>   Entry at address: 0x40194000, end: 0x40198000, level: 2
[    4.348734] ERR: [libukallocbbuddy] <bbuddy.c @ 326>   Entry at address: 0x40014000, end: 0x40018000, level: 2
[    4.349433] ERR: [libukallocbbuddy] <bbuddy.c @ 322> Free list 3 (Order 3):
[    4.349767] ERR: [libukallocbbuddy] <bbuddy.c @ 326>   Entry at address: 0x44108000, end: 0x44110000, level: 3
[    4.350203] ERR: [libukallocbbuddy] <bbuddy.c @ 326>   Entry at address: 0x40198000, end: 0x401a0000, level: 3
[    4.350842] ERR: [libukallocbbuddy] <bbuddy.c @ 326>   Entry at address: 0x40018000, end: 0x40020000, level: 3
```

```
[    4.285120] ERR: [libukallocbbuddy] <bbuddy.c @ 322> Free list 4 (Order 4):
[    4.285793] ERR: [libukallocbbuddy] <bbuddy.c @ 326>   Entry at address: 0x44110000, end: 0x44120000, level: 4
[    4.286666] ERR: [libukallocbbuddy] <bbuddy.c @ 320> Free list 5 is empty.
[    4.287190] ERR: [libukallocbbuddy] <bbuddy.c @ 322> Free list 6 (Order 6):
[    4.287674] ERR: [libukallocbbuddy] <bbuddy.c @ 326>   Entry at address: 0x44140000, end: 0x44180000, level: 6
[    4.288478] ERR: [libukallocbbuddy] <bbuddy.c @ 326>   Entry at address: 0x401c0000, end: 0x40200000, level: 6
[    4.289307] ERR: [libukallocbbuddy] <bbuddy.c @ 326>   Entry at address: 0x40040000, end: 0x40080000, level: 6
[    4.290069] ERR: [libukallocbbuddy] <bbuddy.c @ 322> Free list 7 (Order 7):
[    4.290377] ERR: [libukallocbbuddy] <bbuddy.c @ 326>   Entry at address: 0x44180000, end: 0x44200000, level: 7
[    4.290825] ERR: [libukallocbbuddy] <bbuddy.c @ 326>   Entry at address: 0x40080000, end: 0x40100000, level: 7
[    4.291859] ERR: [libukallocbbuddy] <bbuddy.c @ 320> Free list 8 is empty.
[    4.292536] ERR: [libukallocbbuddy] <bbuddy.c @
```

```
[    4.351791] ERR: [libukallocbbuddy] <bbuddy.c @ 322> Free list 4 (Order 4):
[    4.352292] ERR: [libukallocbbuddy] <bbuddy.c @ 326>   Entry at address: 0x44110000, end: 0x44120000, level: 4
[    4.353012] ERR: [libukallocbbuddy] <bbuddy.c @ 320> Free list 5 is empty.
[    4.353387] ERR: [libukallocbbuddy] <bbuddy.c @ 322> Free list 6 (Order 6):
[    4.353915] ERR: [libukallocbbuddy] <bbuddy.c @ 326>   Entry at address: 0x44140000, end: 0x44180000, level: 6
[    4.354618] ERR: [libukallocbbuddy] <bbuddy.c @ 326>   Entry at address: 0x401c0000, end: 0x40200000, level: 6
[    4.355062] ERR: [libukallocbbuddy] <bbuddy.c @ 326>   Entry at address: 0x40040000, end: 0x40080000, level: 6
[    4.355499] ERR: [libukallocbbuddy] <bbuddy.c @ 322> Free list 7 (Order 7):
[    4.356192] ERR: [libukallocbbuddy] <bbuddy.c @ 326>   Entry at address: 0x44180000, end: 0x44200000, level: 7
[    4.357264] ERR: [libukallocbbuddy] <bbuddy.c @ 326>   Entry at address: 0x40080000, end: 0x40100000, level: 7
[    4.358048] ERR: [libukallocbbuddy] <bbuddy.c @ 320> Free list 8 is empty.
[    4.358623] ERR: [libukallocbbuddy] <bbuddy.c @
```

```
322> Free list 9 (Order 9):
[    4.293136] ERR:
[libukallocbbuddy] <bbuddy.c @
326>   Entry at address:
0x44200000, end: 0x44400000,
level: 9
[    4.293952] ERR:
[libukallocbbuddy] <bbuddy.c @
326>   Entry at address:
0x40200000, end: 0x40400000,
level: 9
```

```
322> Free list 9 (Order 9):
[    4.359185] ERR:
[libukallocbbuddy] <bbuddy.c @
326>   Entry at address:
0x44200000, end: 0x44400000,
level: 9
[    4.359941] ERR:
[libukallocbbuddy] <bbuddy.c @
326>   Entry at address:
0x40200000, end: 0x40400000,
level: 9
```