



北京航空航天大学  
BEIHANG UNIVERSITY

## 操作系统 Lab2 - 1 课上测试简介



- 考试时间 14:15 ~ 16: 00
- 每次课上测试题目分为基础测试和附加测试(选做)两部分
- 通过课上测试的条件是基础测试通过（基础题成绩 $\geq 60$ ）
- 通过附加测试将会给予额外加分（附加题成绩 $\geq 60$ ）
- 课上测试会对课下内容进行强测
- 注意，若未通过基础测试，则无论是否通过附加测试均记为未通过。



## Lab2课上基础题



## Step1:创建lab2-1-exam分支

- `cd ~/学号-lab/`
- `git checkout lab2`
- `git add .`
- `git commit -m "xxxxx"`(填写修改或可提示自己的信息)
- (若lab2分支无改动以上两步可省略)
- `git checkout -b lab2-1-exam` (有参数-b)



## Step2:完成lab2基础题代码编写



## Step3:提交更改

- `cd ~/学号-lab/`
- `git add .`
- `git commit -m "xxxxx"`(填写修改或可提示自己的信息)
- `git push origin lab2-1-exam:lab2-1-exam`



## Step4:提交结果

```
End build at Wed Mar 24 22:33:44 CST 2021  
[ PASSED:55 ]  
[ TOTAL:55 ]  
Begin build at Wed Mar 24 22:33:54 CST 2021  
Cleaning the project
```

```
End build at Wed Mar 24 22:33:56 CST 2021  
[ PASSED:2 ]  
[ TOTAL:2 ]  
[ You got 100 (of 100) this time. Wed Mar 24 22:34:06 CST 2021 ]
```

正确完成实验要求后提交代码，可以看到如上图所示两部分评测结果，得到100分。

此时lab2-1-exam测试通过（**大于60即为通过**），可以选做Extra部分。



### 题目背景

我们实现的MOS操作系统中，所有的物理页的可能状态有三种：使用中物理页、空闲物理页、已经被申请但未被使用的物理页。

Note: `page_alloc`的时候只是申请了一个物理页，但是物理页没有使用，请仔细思考这三种状态物理页的判定方法。

说明：

使用中的物理页：当前使用次数不为0的物理页，状态标记为1

已经被申请但未使用的物理页：当前使用次数为0，但是已经被申请出去的物理页，状态标记为2

空闲物理页：当前可以被申请的物理页，状态标记为3





### 任务1:

在pmap.c中实现函数int page\_alloc2(struct Page \*\*pp), 并在pmap.h中添加该函数的声明.其功能与原有的page\_alloc完全一样 (你可以直接复制page\_alloc的代码), 唯一的区别在于, 如果确实分配到了物理页面, 该函数要输出分配到的物理页的信息  
输出格式: `printf("page number is %x, start from pa %x\n",ppn,pa);`  
其中ppn为页号, pa为该页面的起始物理地址

### 任务2:

在pmap.c中实现函数 void get\_page\_status(int pa); 并在pmap.h中添加该函数的声明。函数输入的是一个物理地址, 请按格式输出该物理页的状态信息。

输出格式: `printf("times:%d, page status:%d\n",var1,var2);`

其中var1是统计该函数被调用的次数 (首次从1开始) , var2是返回该物理地址对应的页面状态标记数字。

**评测要求: 请确保page\_init初始化后page\_free\_list从表头到表尾物理页下标依次递减**

### 任务3:

本次课上测试会对课下测试进行加强测试, 请大家在pmap.h中添加以下函数定义

(请不要在pmap.c中添加这两个函数的实现, 否则远端测评无法编译):

```
void test_queue();
```

```
void pm_check();
```



## Lab2-1课上附加题



## Step5:创建附加题分支（选做）

- `git checkout lab2` （回到lab2分支下）
- `git add .`
- `git commit -m "xxxxx"` (填写修改或可提示自己的信息)
- (若lab2分支无改动以上两步可省略)
- `git checkout -b lab2-1-Extra`



## Step6:完成lab2-1附加题代码编写（选做）



## Step7:提交更改（选做）

- 必须通过基础测试才能获得附加题分数
- `cd ~/学号-lab/`
- `git add .`
- `git commit -m "xxxxx"`(填写修改或可提示自己的信息)
- `git push origin lab2-1-Extra:lab2-1-Extra`



## Step8:提交结果（选做）

```
End build at Mon Apr 5 23:49:09 CST 2021  
[ Start test. ]  
[ PASSED:9 ]  
[ TOTAL:9 ]  
[ You got 100 (of 100) this time. Mon Apr 5 23:49:19 CST 2021 ]
```

正确完成实验要求后提交代码，可以看到如上图所示评测结果，得到100分。

此时lab2-1-Extra测试通过，在完成lab2-1-exam的情况下，可以获得lab2-1课上测试的额外加分。



## Lab2-1-Extra测试题目

### 题目背景：

常见的管理空闲内存的方法有两种：链表管理法与位图管理法。在lab2中，我们通过链表实现了对空闲内存的管理，下面的extra部分将要求大家实现用位图管理空闲内存。

用位图管理内存需要给每个分配单元赋予一个**二进制数位**，用来记录该分配单元是否闲置。

### 位图规格要求：

1. 数位取值为0表示单元闲置，取值为1则表示已被占用。
2. 用一个**unsigned int page\_bitmap**数组管理内存，要求在该数组中，标号小的元素的低位表示页号小的页面。例如，0号页面由page\_bitmap[0]的第0位表示，63号页面由page\_bitmap[1]的第31位表示。当只有0号页面与63号页面被占用时，应该有：page\_bitmap[0]=0x00000001, page\_bitmap[1]=0x80000000



## Lab2-1-Extra测试题目

### 题目要求:

**任务一:** 在pmap.c中添加如下的空闲页面位图定义:

```
unsigned int page_bitmap[NUM];
```

其中NUM是一个你需要计算的数, 要求这个数组可以恰好表示所有物理页面, 不多不少。

修改page\_init(), 除需要初始化位图外, 需要添加输出:

```
printf("page bitmap size is %x\n", NUM);
```

其中NUM为page\_bitmap数组的元素个数

### 任务二:

修改page\_alloc(struct Page \*\*pp), 要求分配到的页面是空闲页面中页号最小的

### 任务三:

修改page\_free(struct Page \*pp)

三个函数修改后需要满足前述的位图规格要求。除页面组织形式外, 其他要求与课下要求相同。

**注意:** 请保证没有使用链表相关操作组织页面, 评测时若发现使用链表组织页面将不予通过!

**评测要求:** 为了正确评测, 请在pmap.h中添加以下函数定义

(请不要在pmap.c中添加这个函数的实现, 否则远端测评无法编译):

```
void pm_check(void);
```





北京航空航天大学  
BEIHANG UNIVERSITY

下面请同学们开始做题  
有问题可以随时提问

祝实验顺利！