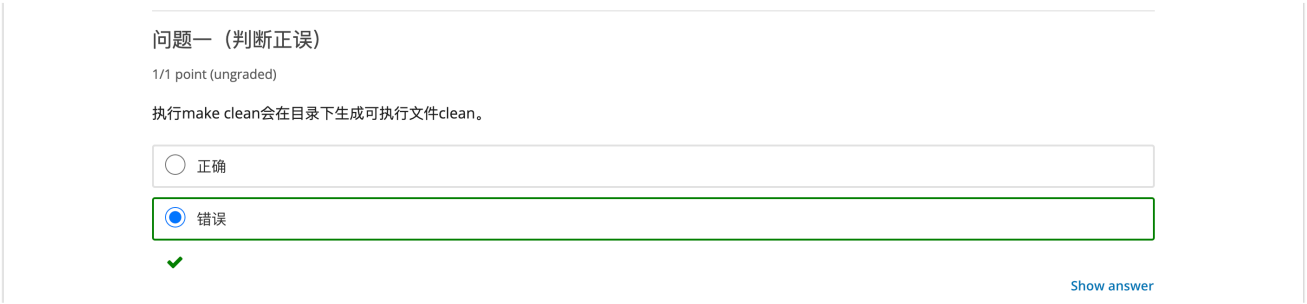
Lab1

1. 课程 > 内核、Boot 和 printf > 操作系统的启动 > 内核在哪里?

2. 课程 > 内核、Boot 和 printf > 操作系统的启动 > Bootloader

3. 课程 内核、Boot 和 printf Let's hack the kernel! Makefile—内核代码的地图



4. 课程 > 内核、Boot 和 printf > Let's hack the kernel! > ELF—深入探究编译与链接

问题一（多项选择）

1/1 point (ungraded)

ELF文件包含哪几种文件类型?

- [x] 可重定位文件
- [ ] .h头文件
- [x] 可执行文件
- [x] 共享对象文件

✔

保存 | Show answer

问题二（单项选择）

1/1 point (ungraded)

我们可以使用Elf32_Ehdr结构中的哪部分判断是否是ELF文件?

- (•) e_ident
- ( ) e_type
- ( ) e_machine
- ( ) e_version

✔

保存 | Show answer

问题三（单项选择）

1/1 point (ungraded)

请阅读readelf/kerelf.h，回答：

Elf32_Section的数据类型实际上是?

- ( ) int32_t
- ( ) u_int32_t
- (•) u_int16_t
- ( ) u_int64_t

✔

保存 | Show answer

问题四（单项选择）

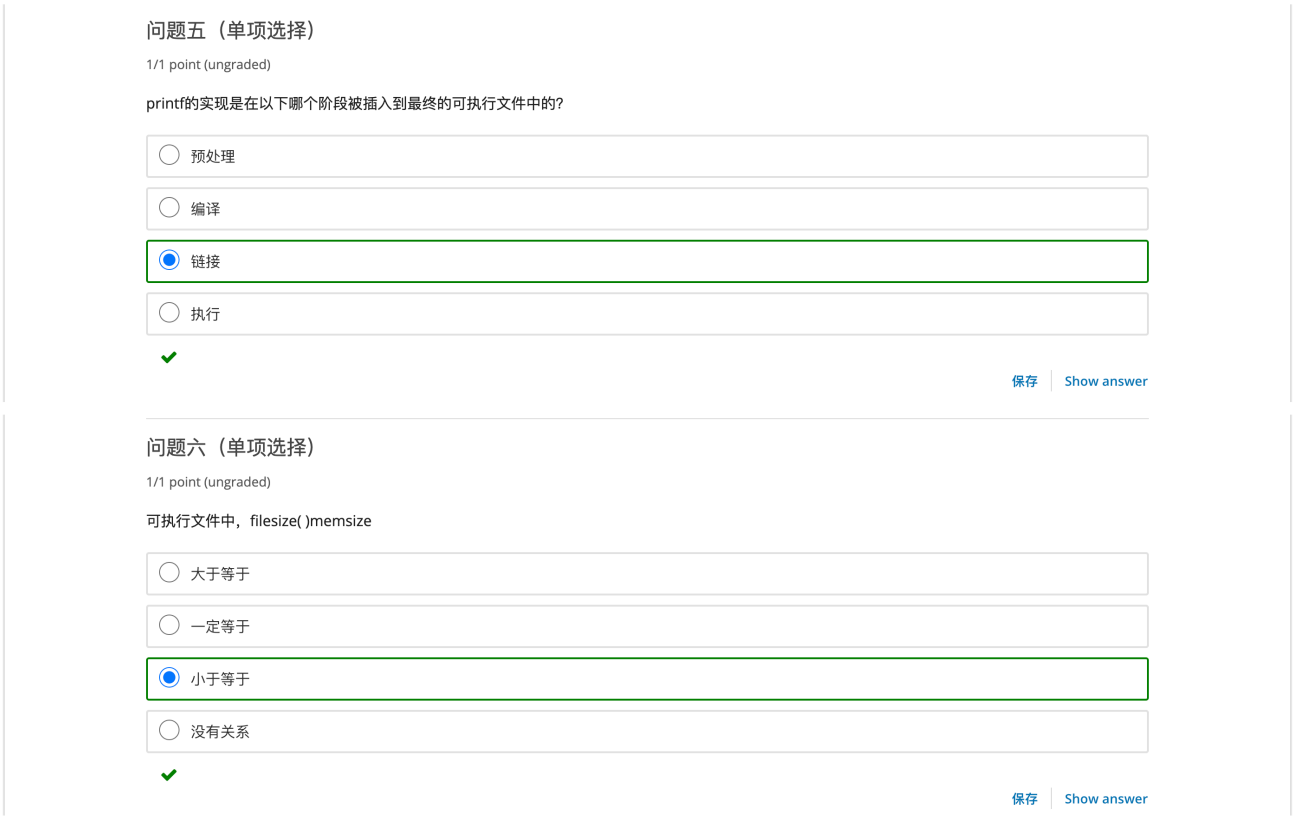
1/1 point (ungraded)

对于可重定位(relocatable)文件，查看其e_type值为多少?

- (•) 1
- ( ) 2
- ( ) 3
- ( ) 4

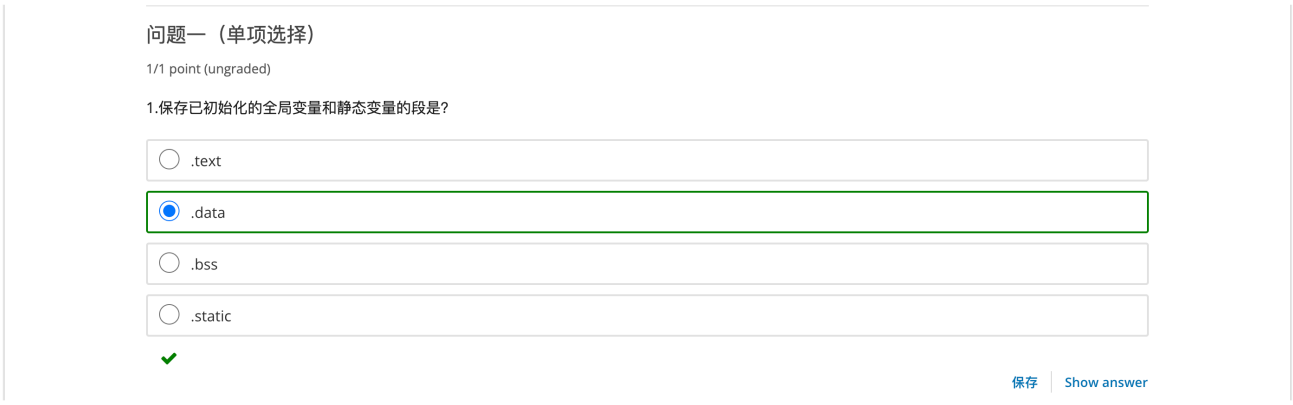
✔

保存 | Show answer



5. 课程 > 内核、Boot 和 printf > Let's hack the kernel! > MIPS内存布局——寻找内核的正确位置

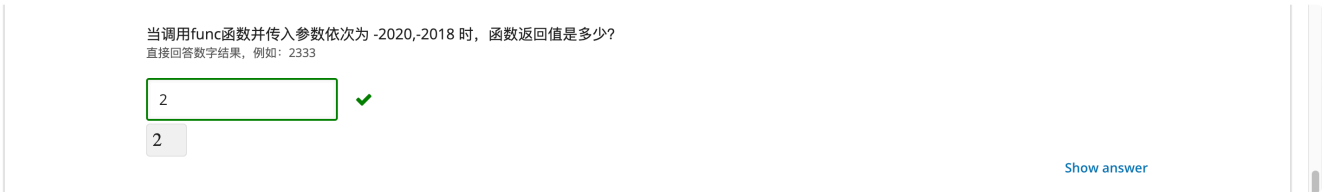
6. 课程 > 内核、Boot 和 printf > Let's hack the kernel! > Linker Script——控制加载地址



7. 课程 > 内核、Boot 和 printf > MIPS汇编与C语言 > 循环与判断

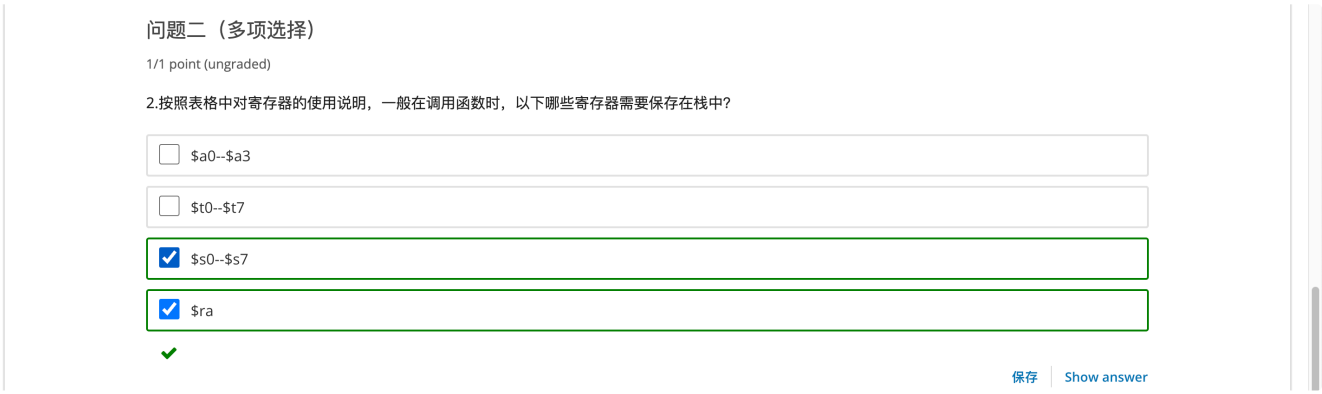

```
...
...
func:
sw $fp, 0($sp)
addi $fp, $sp, 4
sw $ra, -8($fp)
add $t1, $zero, $a0
add $t2, $zero, $a1
bgez $t1, if_else_0
li $t8, 0
sub $t0, $t8, $t1
add $t0, $t0, 0
add $v0, $t2, $t0
move $sp, $fp
lw $fp, -4($sp)
lw $ra, -8($sp)
jr $ra
if_else_0:
if_end_0:
add $t0, $t1, -0
add $v0, $t2, $t0
move $sp, $fp
lw $fp, -4($sp)
```

```
24 lw $ra, -8($sp)
25 jr $ra
26 ...
27 ...
```



8. 课程 > 内核、Boot 和 printf > MIPS汇编与C语言 > 函数调用

9. 课程 > 内核、Boot 和 printf > MIPS汇编与C语言 > 通用寄存器使用约定



0. 课程 > 内核、Boot 和 printf > 实战printf > 实战printf