repo: GitHub

---

int_stack – character-device kernel module

Feature	Implementation
Dynamic memory	`krealloc()` for the integer array
Thread safety	single `mutex` guarding every stack access
file_operations	`open`, `release`, `read (pop)`, `write (push)`, `unlocked_ioctl (set-size)`
IOCTL interface	`_IOW('i', 0, int)`
Error handling	empty → EOF, full → `-ERANGE`, others → `-EINVAL / -ENOTTY / -ENOMEM`

```
/* core snippets */
mutex_lock(&stack.mtx);
if (stack.top == stack.max)           /* push */ return -ERANGE;
if (stack.top == 0)                   /* pop  */ return 0;            /* EOF */
stack.top = 0;                        /* set-size → wipe stack */
mutex_unlock(&stack.mtx);
```

> Since Linux 6.9 `class_create()` takes a single argument, hence `class_create(DEV_NAME)` is used.

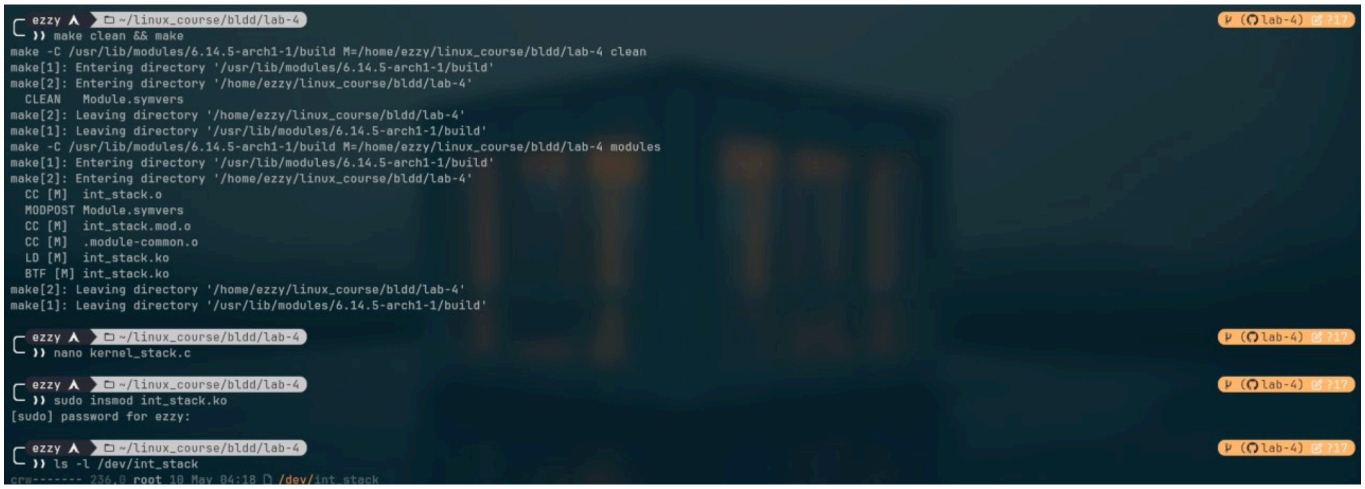
kernel_stack – user-space CLI

- **Commands:** `set-size N`, `push V`, `pop`, `unwind`
- Error format mirrors the task requirements.
- On `-ERANGE` the tool exits with `exit(-ERANGE)` → `$? == 222` (0xDE = −34 mod 256).

```
if (write(fd,&val,sizeof val) < 0 && errno==ERANGE) {
        fputs("ERROR: stack is full\n", stderr);
        exit(-ERANGE);                /* -> 222 */
}
```

Scenario	Result
stack full	`ERROR: stack is full`, `$? = 222`
stack empty	`NULL`, `$? = 0`
`set-size ≤ 0`	`ERROR: size should be > 0`, `$? = 1`

Build & load



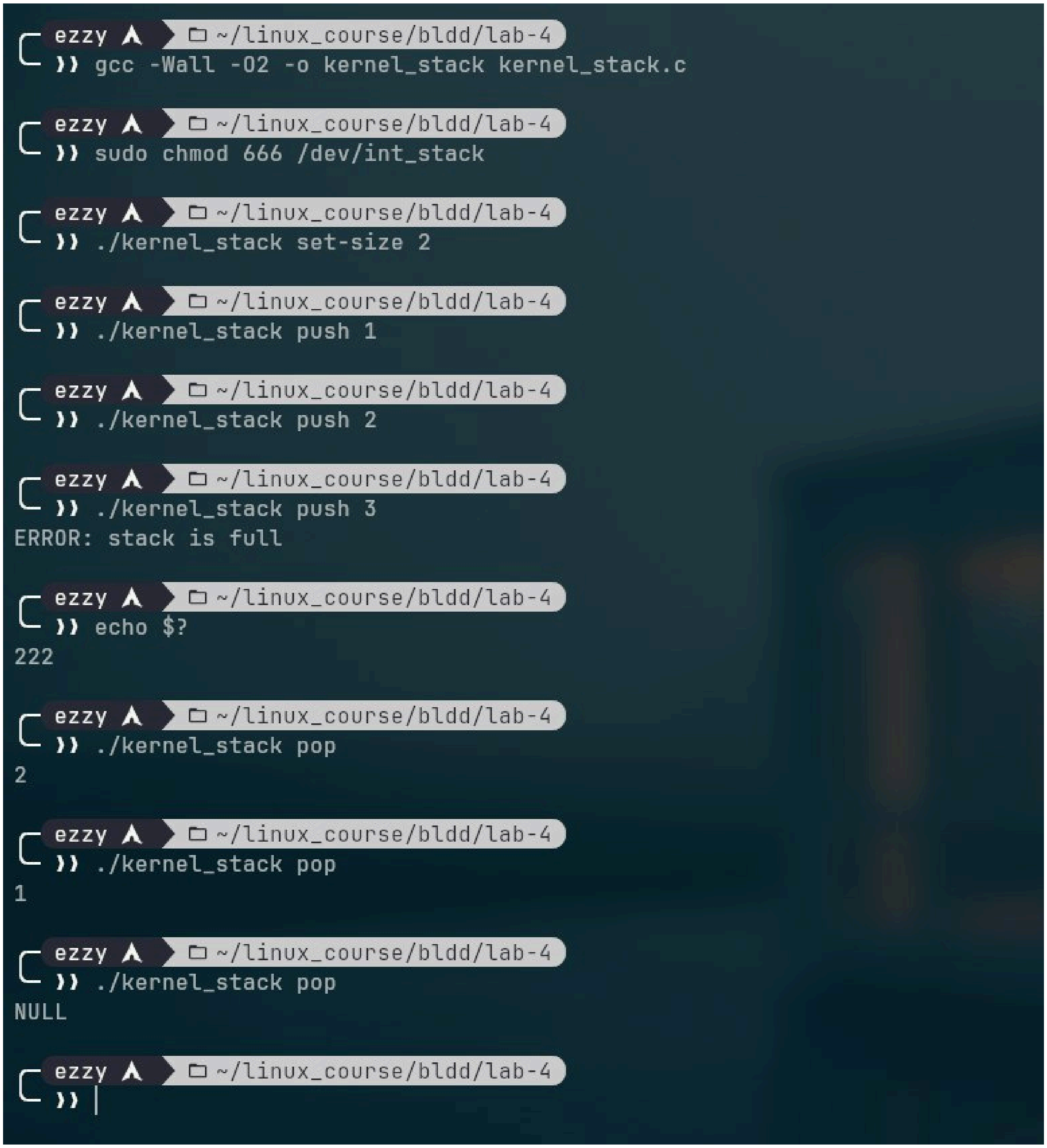
```
ezzy ~/linux_course/bldd/lab-4                                                                      (lab-4) ?17
)) make clean && make
make -C /usr/lib/modules/6.14.5-arch1-1/build M=/home/ezzy/linux_course/bldd/lab-4 clean
make[1]: Entering directory '/usr/lib/modules/6.14.5-arch1-1/build'
make[2]: Entering directory '/home/ezzy/linux_course/bldd/lab-4'
  CLEAN   Module.symvers
make[2]: Leaving directory '/home/ezzy/linux_course/bldd/lab-4'
make[1]: Leaving directory '/usr/lib/modules/6.14.5-arch1-1/build'
make -C /usr/lib/modules/6.14.5-arch1-1/build M=/home/ezzy/linux_course/bldd/lab-4 modules
make[1]: Entering directory '/usr/lib/modules/6.14.5-arch1-1/build'
make[2]: Entering directory '/home/ezzy/linux_course/bldd/lab-4'
  CC [M]  int_stack.o
  MODPOST Module.symvers
  CC [M]  int_stack.mod.o
  CC [M]  .module-common.o
  LD [M]  int_stack.ko
  BTF [M] int_stack.ko
make[2]: Leaving directory '/home/ezzy/linux_course/bldd/lab-4'
make[1]: Leaving directory '/usr/lib/modules/6.14.5-arch1-1/build'

ezzy ~/linux_course/bldd/lab-4                                                                      (lab-4) ?17
)) nano kernel_stack.c

ezzy ~/linux_course/bldd/lab-4                                                                      (lab-4) ?17
)) sudo insmod int_stack.ko
[sudo] password for ezzy:

ezzy ~/linux_course/bldd/lab-4                                                                      (lab-4) ?17
)) ls -l /dev/int_stack
crw------- 236,0 root 10 May 04:18 /dev/int_stack
```

Functional test



```
ezzy  ~/linux_course/bldd/lab-4
)) gcc -Wall -O2 -o kernel_stack kernel_stack.c

ezzy  ~/linux_course/bldd/lab-4
)) sudo chmod 666 /dev/int_stack

ezzy  ~/linux_course/bldd/lab-4
)) ./kernel_stack set-size 2

ezzy  ~/linux_course/bldd/lab-4
)) ./kernel_stack push 1

ezzy  ~/linux_course/bldd/lab-4
)) ./kernel_stack push 2

ezzy  ~/linux_course/bldd/lab-4
)) ./kernel_stack push 3
ERROR: stack is full

ezzy  ~/linux_course/bldd/lab-4
)) echo $?
222

ezzy  ~/linux_course/bldd/lab-4
)) ./kernel_stack pop
2

ezzy  ~/linux_course/bldd/lab-4
)) ./kernel_stack pop
1

ezzy  ~/linux_course/bldd/lab-4
)) ./kernel_stack pop
NULL

ezzy  ~/linux_course/bldd/lab-4
))
```

`/dev/int_stack` permissions are set via a udev rule (`MODE="0666"`).

`unwind` test:

```
./kernel_stack push 1
./kernel_stack push 2
./kernel_stack push 3
ERROR: stack is full
./kernel_stack unwind
2
1
./kernel_stack pop
NULL
```
