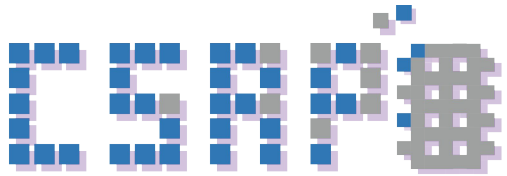
System Programming
Lab Session #3

Kernel Lab Hints

2019/10/08

sysprog@csap.snu.ac.kr



Computer Systems and Platforms Laboratory
School of Computer Science and Engineering
Seoul National University

1. Part A

Part A

- Work in directory `kernellab/paddr`
  - Write a kernel module (implement `dbfs_paddr.c`) that
    - Gets the PID and a virtual address from the user process
    - Returns the physical address of the virtual address

Part A

- When the module loads

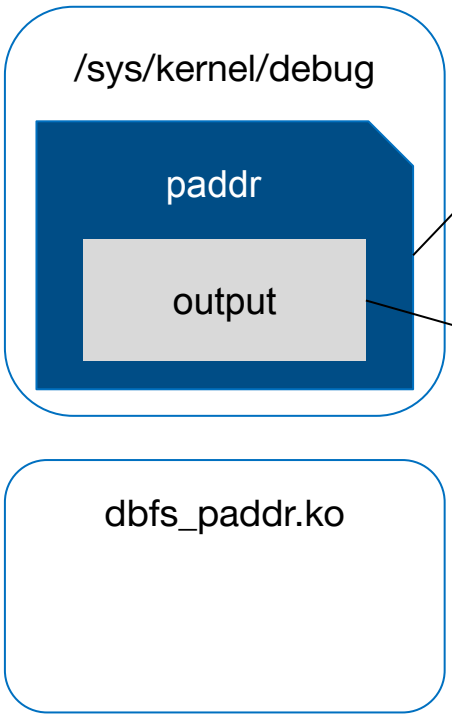
```
static int __init dbfs_module_init(void)
{
        // Implement init module
        printk("Init Module\n");

        dir = debugfs_create_dir("paddr", NULL);

        if (!dir) {
                printk("Cannot create paddr dir\n");
                return -1;
        }

        // Fill in the arguments below
        output = debugfs_create_file("output", 0444, dir, NULL, &dbfs_fops);

        return 0;
}
```

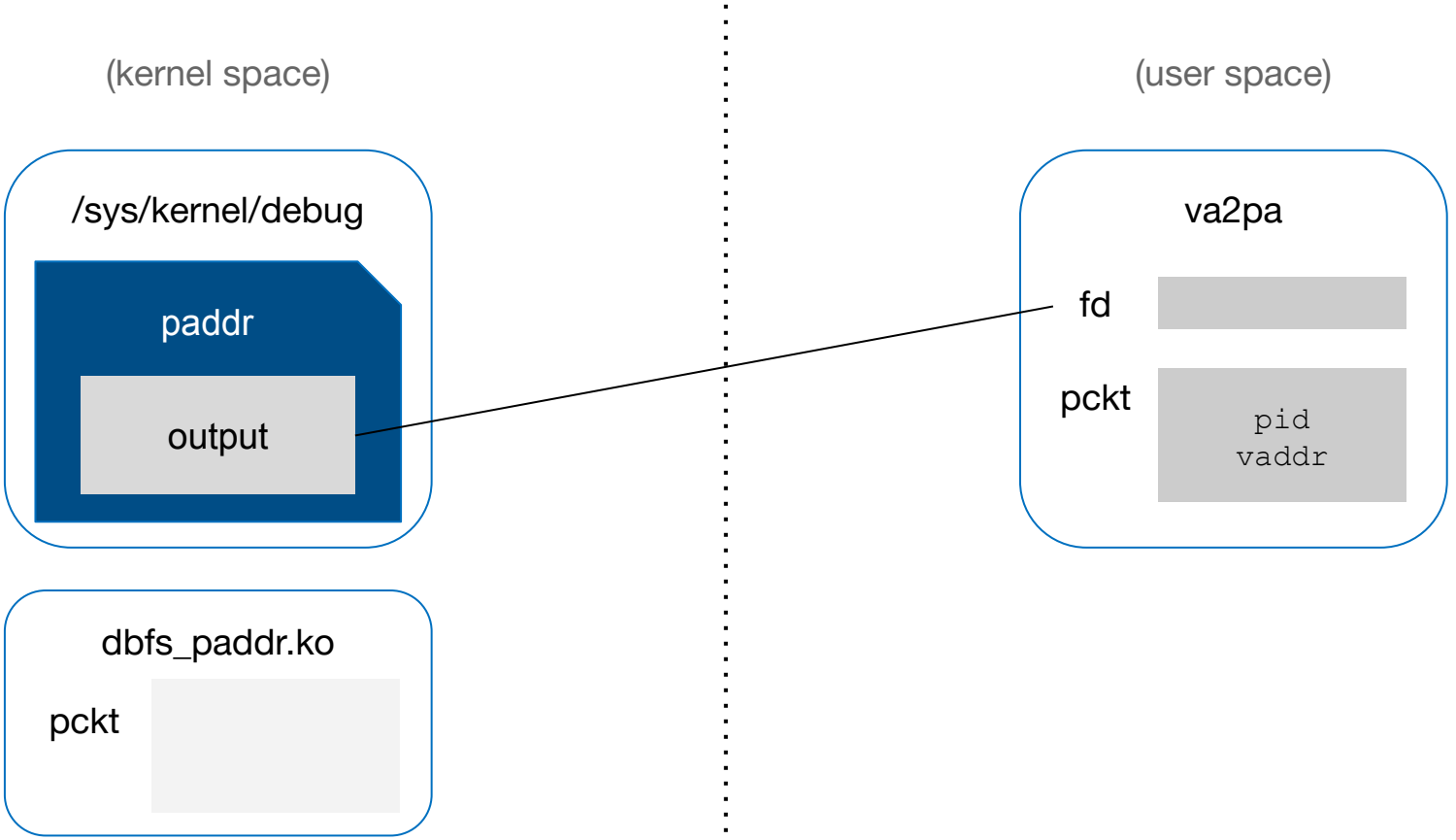


1. directory “paddr” creates under `/sys/kernel/debug`

2. file “output” creates under `/sys/kernel/debug/paddr`

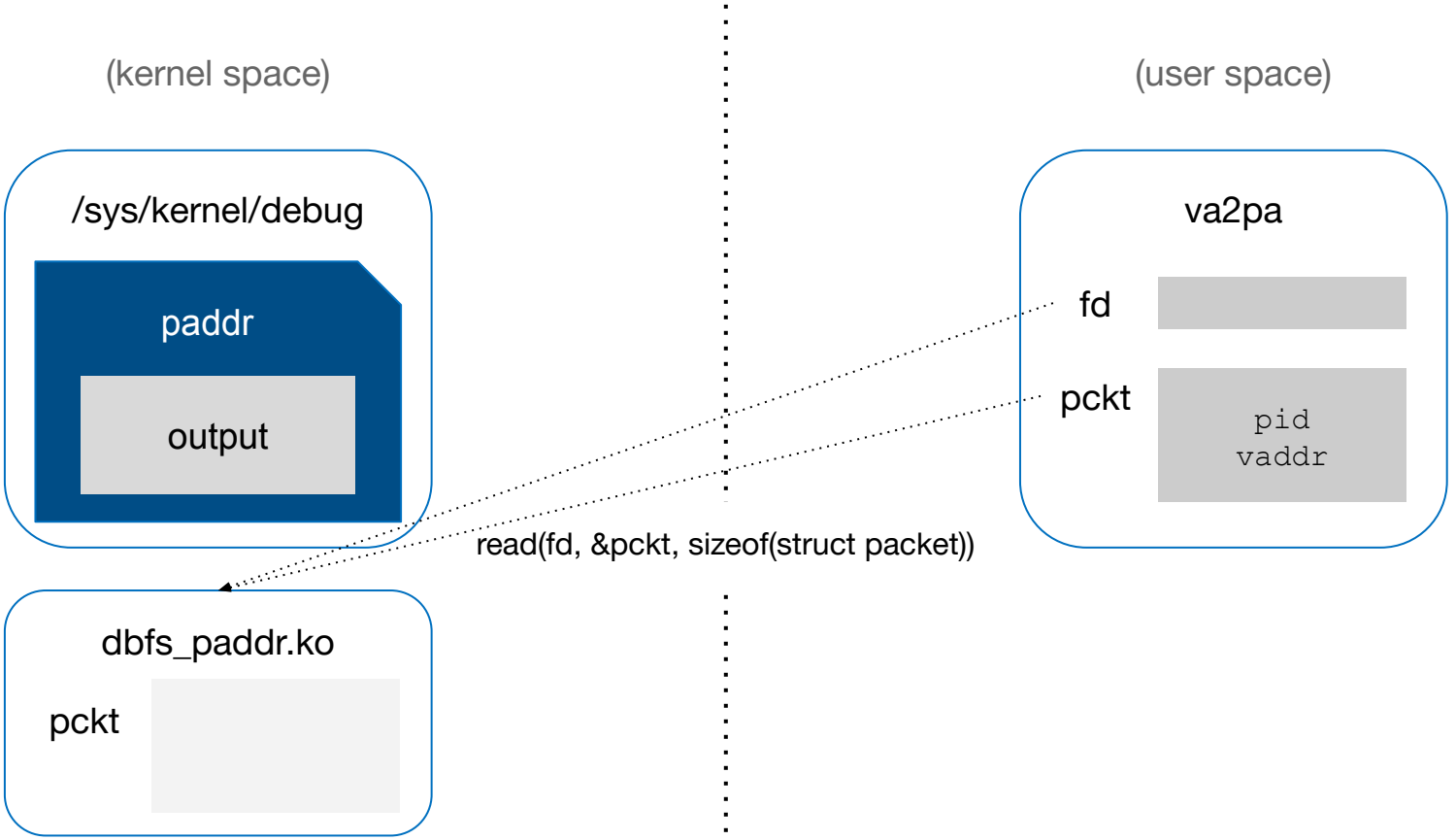
Part A

- By executing the program *va2pa*



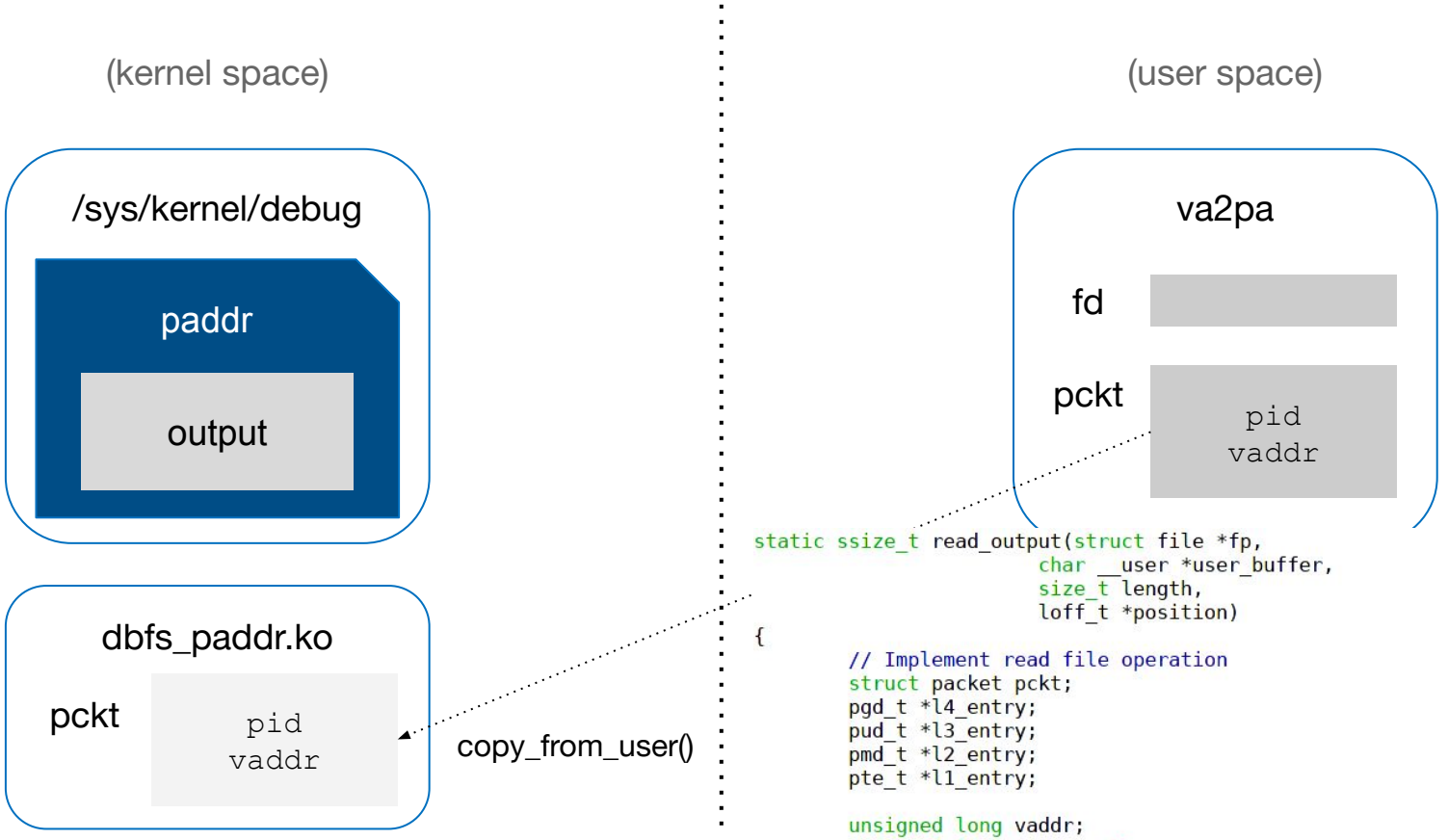
Part A

- By executing the program *va2pa*



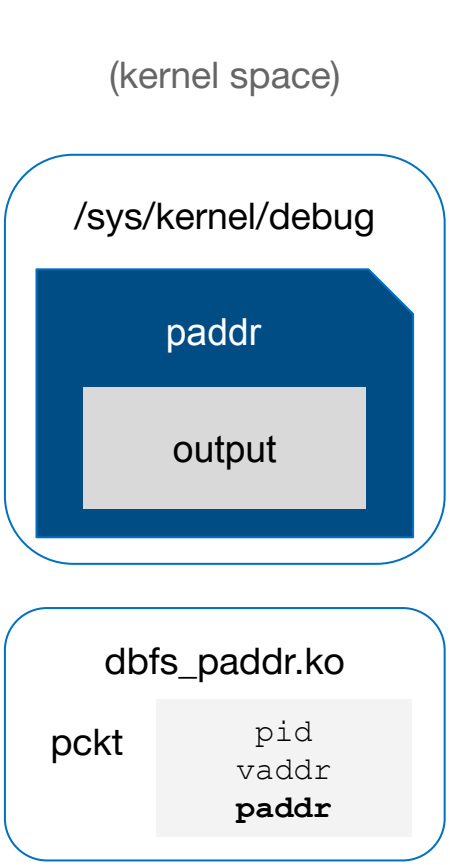
Part A

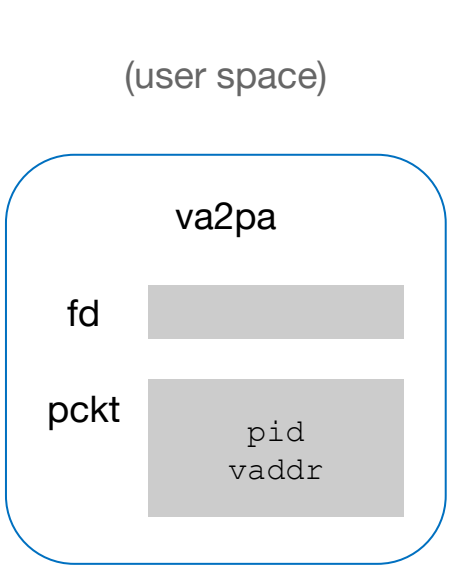
- By executing the program *va2pa*



Part A

- By executing the program *va2pa*



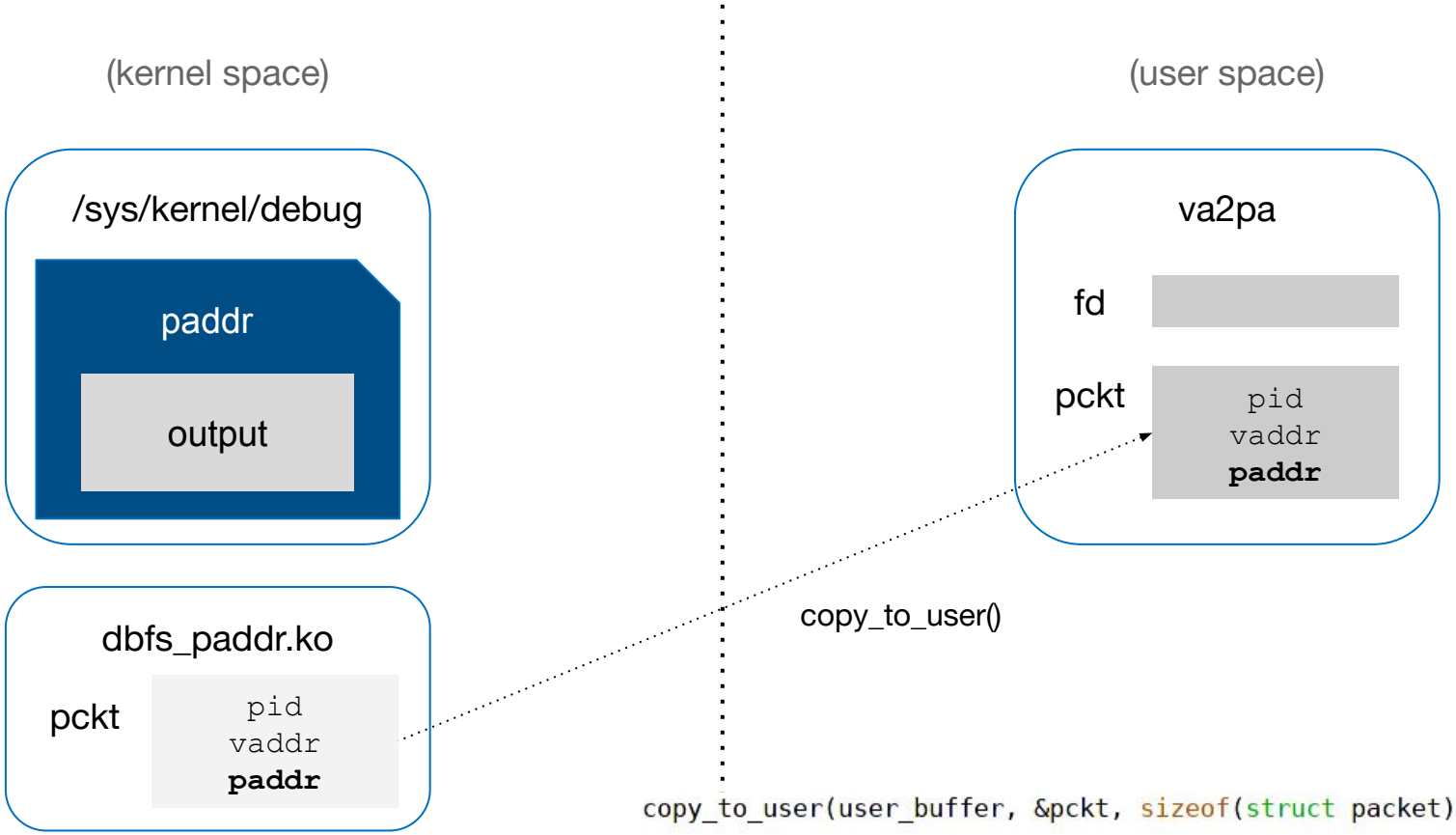



```
task = pid_task(find_vpid(pckt.pid), PIDTYPE_PID);
vaddr = pckt.vaddr;
l4_entry = pgd_offset(task->mm, vaddr);
```

l3_entry = ...

Part A

- By executing the program *va2pa*




```
copy_to_user(user_buffer, &pckt, sizeof(struct packet));

return length;
}
```

Part A

- When the module unloads

/sys/kernel/debug

```
static void __exit dbfs_module_exit(void)
{
        // Implement exit module
        debugfs_remove_recursive(dir);
        printk("Exit Module\n");
}
```

2. Part B

Part B

- Work in directory `kernellab/ptrav`
  - Write a kernel module (implement `dbfs_ptrav.c`) that
    - Gets the PID of the program *test* from the *rss*
    - Returns the resident set size (rss) of *test*