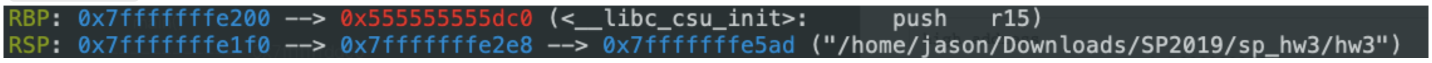
System Programming HW3 Report

B04901003 許傑盛

(a) Draw the stack frame

```
high address
        --------------------------------------
                        rbp = 0x7fffffffe200
            main()      rsp = 0x7fffffffe1f0
        --------------------------------------
                        rbp = 0x7fffffffe1e0
            dummy()     rsp = 0x7fffffff4580
        --------------------------------------
                        rbp = 0x7fffffff4570
            funct_1()   rsp = 0x7fffffff4540
        --------------------------------------
                        rbp = 0x7fffffff4530
            dummy()     rsp = 0x7ffffffea8d0
        --------------------------------------
                        rbp = 0x7ffffffea8c0
            funct_2()   rsp = 0x7ffffffea890
        --------------------------------------
                        rbp = 0x7ffffffea880
            dummy()     rsp = 0x7ffffffe0c20
        --------------------------------------
                        rbp = 0x7ffffffe0c10
            funct_3()   rsp = 0x7ffffffe0be0
        --------------------------------------
                        rbp = 0x7ffffffe0bd0
            dummy()     rsp = 0x7ffffffd6f70
        --------------------------------------
                        rbp = 0x7ffffffd6f60
            funct_4()   rsp = 0x7ffffffd6f30
        --------------------------------------
low address
```

`main()` : rsp, rbp



```
RBP: 0x7fffffffe200 --> 0x555555555dc0 (<__libc_csu_init>:      push   r15)
RSP: 0x7fffffffe1f0 --> 0x7fffffffe2e8 --> 0x7fffffffe5ad ("/home/jason/Downloads/SP2019/sp_hw3/hw3")
```

(b) local variable

Since the variables stored in stack memory weren't changed before jump back to the same function. When program continued to execute the function, CPU read out the variable value from stack memory, and thus remain the same.

(c) usage of the dummy function

Without dummy function, if there is some local variables inside the signal handler or scheduler, it may changed the content of the stack memory and thus changed the stored variables in another function. When jump to that function, it may have some undefined outcome.

(d) switch to funct_4 and call return in funct_4

The program would first return from `funct_4()` and return from `dummy()` . However, after that the program just continue executing the line after call `dummy()` in `funct_3()` , and there would be another jump to scheduler but not return.

(e) how do you finish your program

I didn't do anything special, just carefully read and follow the spec to finish the this homework.