System Programming HW3 report b07902129 何政勳

A.

| Stack frame | Base pointer | Stack pointer |
|---|---|---|
| Main | 0x7fffffffdc50 | 0x7fffffffda60 |
| dummy | 0x7fffffffda50 | 0x7fffffff3df0 |
| funct_1 | 0x7fffffff3de0 | 0x7fffffff3dc0 |
| dummy | 0x7fffffff3db0 | 0x7ffffffea150 |
| funct_2 | 0x7ffffffea140 | 0x7ffffffea120 |
| dummy | 0x7ffffffea110 | 0x7ffffffe04b0 |
| funct_3 | 0x7ffffffe04a0 | 0x7ffffffe0480 |
| dummy | 0x7ffffffe0470 | 0x7ffffffd6810 |
| funct_4 | 0x7ffffffd6800 | 0x7ffffffd67e0 |

use gdb to find the address of each function

1. gcc -g hw3.c -o hw3
2. gdb hw3
3. b 95 //set breakpoint just before the long jump to main
4. r 5 3 1 0 //run hw3 with arguments 5 3 1 0
5. frame [0-8] //select a frame indexing from 0 to 8
6. info frame // show the information of the selected frame
7. info reg // show the contents of registers of the selected frame

B. Yes, setjmp saves the stack pointer of the environment, when longjmp is called, it can restore (rewind) the stack to the previous status. If a variable is saved in stack memory and the stack memory isn't damaged by other function, then the variable can be restored.

C. Dummy function is used to separate all the others function to prevent damage on the memory of stack frames. For example, if there wasn't a dummy function, after calling the scheduler function at the first time in main function, the stack frame of funct_1 would be damaged.

D. No, it can't. The stack frames of dummy, funct_3, dummy … aren't restored. In gdb I set up two break points, the first one is right before funct_4's return, the second one is right before dummy's return. The funct_4 ones showed up correctly, however, before the breakpoint in dummy showed up, an error "stack smashing detected" came up. Also the process received a SIGABRT.

E. I finished my program mainly by checking up course slides, Linux programmer's manual and discussion with b07902125. One of the hardest part is understanding the task 3 since there are much more things to do than task 1 and 2. The queue in task 3 is kind of confusing, which takes me some time to figure out what it is. Also sigaction is used since signal is unsafe and system depending.