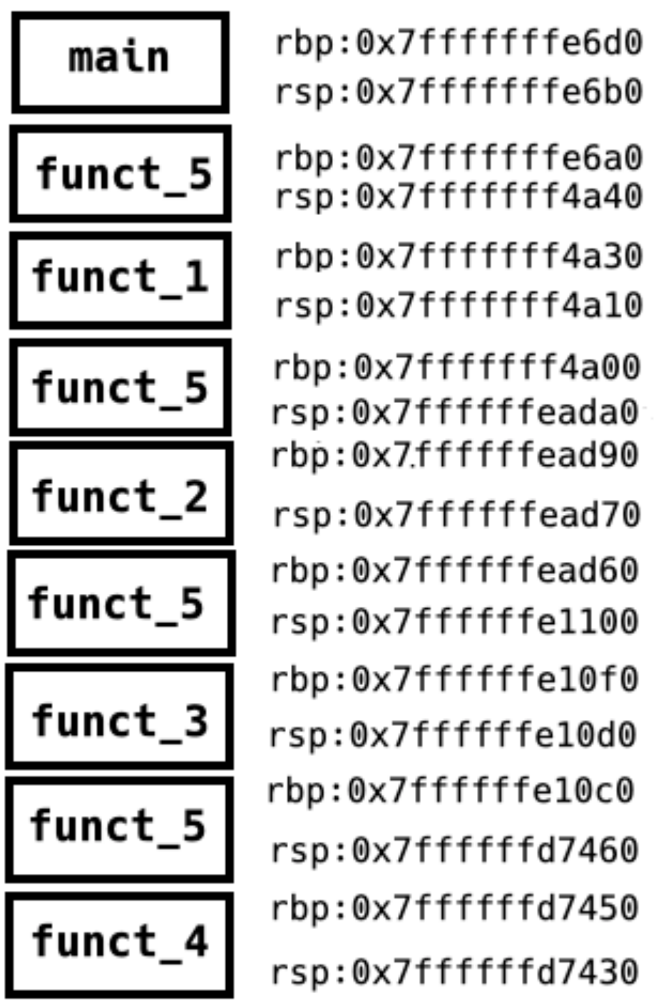
System Programming HW3 report

b07902064 資工二 蔡銘軒

(a)



(b) Yes. The stack frame of the function is unchanged. We have used the dummy as a buffer to avoid functions from overwriting the stack frame of other functions. So the value is unaffected by "setjmp" and "longjmp".

(c) The dummy function acts as a buffer between the functions. We call "Scheduler()" and the kernel calls the signal handler on our behalf in our functions. They are pushed into the stack on top of our functions. If there is no buffer between our functions, they may overlap with the stack frame of other functions. When we later use "longjmp()" to return to a function, the content may already be corrupted.

(d) No. In fact, when I tested it on the workstation, it resulted in a "stack smashing". When I used GDB to see what was happening underneath the hood, I found that the "stack smashing" occurred when "funct_5" was trying to return to "main". I suspect the reason had something to do with the "rip" register. When "main" first called "funct_5", "funct_5" was pushed onto the stack. In the stack frame, various information was stored, including the "rip" register, which stored the instruction to execute after the function returned. Later when "funct_4" long jumped back to "main", "main" called the "Scheduler" function. The new stack frame overlapped with the original stack frame of "funct_5", overwriting the information, including the content of "rip" register. When "funct_5" was trying to return to "main", the system(Stack Smashing detector) detected the stack frame was altered and it was returning to some place it was not supposed to return to. So it raised a SIGABRT to terminate the process.

(e) I read the textbook to make sure I had enough understanding of "setjmp()" and "longjmp()" before I started writing the code. All the functions used were taught in class and thus I didn't struggle with it. Most of the time, I was trying to understand how the given "scheduler.o" and "scheduler.c" worked. I didn't know much about compiling multiple files into one executable file, so it took me a while to figured it out.