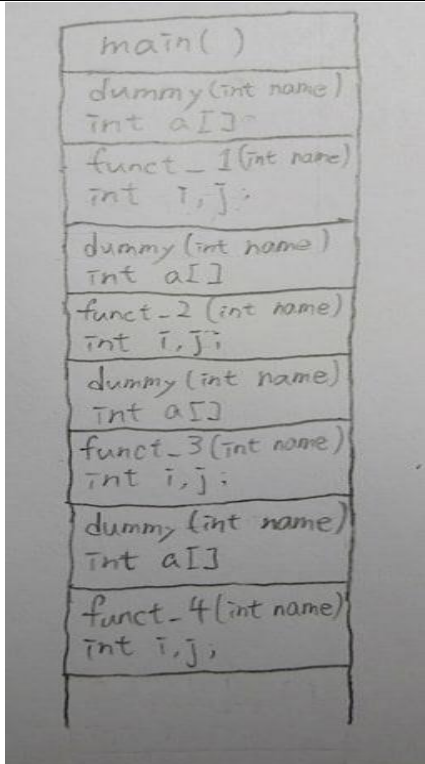


學號:B06902136

系級:資工三

姓名:賴冠毓

a.

Stack frame	用 gdb 的結果如下: (不含 main)
	Breakpoint 5, funct_5 (name=1) at hw3.c:175 175 { (gdb) info reg rbp 0x7fffffff760 0x7fffffff760 rsp 0x7fffffff4b00 0x7fffffff4b00
	Breakpoint 1, funct_1 (name=1) at hw3.c:62 62 if(setjmp(block[1]->Environment) == 0) (gdb) info reg rbp 0x7fffffff4af0 0x7fffffff4af0 rsp 0x7fffffff4ad0 0x7fffffff4ad0
	Breakpoint 5, funct_5 (name=2) at hw3.c:175 175 { (gdb) info reg rbp 0x7fffffff4ac0 0x7fffffff4ac0 rsp 0x7ffffffeae60 0x7ffffffeae60
	Breakpoint 2, funct_2 (name=2) at hw3.c:91 91 if(setjmp(block[2]->Environment) == 0) (gdb) info reg rbp 0x7ffffffeae50 0x7ffffffeae50 rsp 0x7ffffffeae30 0x7ffffffeae30
	Breakpoint 5, funct_5 (name=3) at hw3.c:175 175 { (gdb) info reg rbp 0x7ffffffeae20 0x7ffffffeae20 rsp 0x7ffffffellc0 0x7ffffffellc0
	Breakpoint 3, funct_3 (name=3) at hw3.c:120 120 if(setjmp(block[3]->Environment) == 0) (gdb) info reg rbp 0x7ffffffellb0 0x7ffffffellb0 rsp 0x7ffffffell90 0x7ffffffell90
	Breakpoint 5, funct_5 (name=4) at hw3.c:175 175 { (gdb) info reg rbp 0x7ffffffell80 0x7ffffffell80 rsp 0x7ffffffd7520 0x7ffffffd7520
	Breakpoint 4, funct_4 (name=4) at hw3.c:149 149 if(setjmp(block[4]->Environment) == 0) (gdb) info reg rbp 0x7ffffffd7510 0x7ffffffd7510 rsp 0x7ffffffd74f0 0x7ffffffd74f0

b.

Yes, local variable 的值會一樣。

因為 local variable 是存在每個 function 自己的 stack frame，而只要這個 function 沒有 return 或 stack frame 沒有被蓋掉(dummy 的功能)，那 jump 回來 local variable 還是保持上次 jump 離開時的值。

c.

dummy 的功能就是避免 funct_1 的 stack frame 被 scheduler 覆蓋掉。如果沒有 dummy，整個 stack frame 就是 main -> funct_1 -> funct_2 -> funct_3 -> funct_4，那 initialize 後 jump 回 main，會讓 jump_buf 回到 main 的 stack frame，再來跑 Scheduler() 就會把 funct_1 的 stack frame 被 Scheduler 的 stack frame 蓋掉。

d.

No，在 main 的前一個 dummy 已經被 Scheduler() 取代了，所以回不去了。

用 gdb 的結果如下：

```

(gdb) b Scheduler
Breakpoint 1 at 0x400909
(gdb) r 1 1 1 0
Starting program: /mnt/d/download/test 1 1 1 0
Breakpoint 1, 0x000000000400909 in Scheduler ()
(gdb) n
Single stepping until exit from function Scheduler,
which has no line number information.
child switch to funct_4
0x000000000400edb in funct_4 (name=4) at test.c:187
187         if(setjmp(block[4]->Environment) == 0)
(gdb) n
193         return;
(gdb) n
214     }
(gdb) n
funct_5 (name=4) at test.c:236
236     }(gdb) n
funct_3 (name=3) at test.c:181
181     }
(gdb) n
funct_5 (name=3) at test.c:236
236     }(gdb) n
funct_2 (name=2) at test.c:149
149     }
(gdb) n
funct_5 (name=2) at test.c:236
236     }(gdb) n
funct_1 (name=1) at test.c:117
117     }
(gdb) n
funct_5 (name=1) at test.c:236
236     }(gdb) n
*** stack smashing detected ***: /mnt/d/download/test terminated

Program received signal SIGABRT, Aborted.
0x00007ffff065428 in __GI_raise (sig=sig@entry=6) at ../sysdeps/unix/sysv/linux/raise.c:54
54 ..
../sysdeps/unix/sysv/linux/raise.c: No such file or directory.

```

e.

How I finish the program:

一開始在建 linked list 有 bug 時，有參考這邊去做修改才成功：

[https://github.com/hasancse91/data-structures/blob/master/Source%20Code/Circular%20Doubly%20Linked%20List%20\(Inser%20Delete%20Print\).c?fbclid=IwAR1d6tC0jIcaScQT7FDAF_pJSKC-QKnB2MhODv66itBA5-ppQUFQBSTuVvY](https://github.com/hasancse91/data-structures/blob/master/Source%20Code/Circular%20Doubly%20Linked%20List%20(Inser%20Delete%20Print).c?fbclid=IwAR1d6tC0jIcaScQT7FDAF_pJSKC-QKnB2MhODv66itBA5-ppQUFQBSTuVvY)

寫 report 有些 stack frame 概念不太清楚，看這個影片後才搞懂：

https://www.youtube.com/watch?v=WGxZ5SQ4kvA&fbclid=IwAR3NgDgmYSz6YousEU09CXRST0xNMfBeWfISMC-iwyOtZUKZd_sAHhADwrM

剩下的部分照著 spec 操作就成功了。

Special Thanks to:

(1)在 e-mail 中回答我問題的 TA 們。

(2)B07902144: 一開始 segmentation fault 跟寫 report 時有跟他討論，才順利找出 bug 跟釐清觀念，還有使用 gdb 有遇到很多奇怪的狀況，有請他幫忙看看是哪裡弄錯了，最後才順利寫好 report 要用 gdb 的部分。