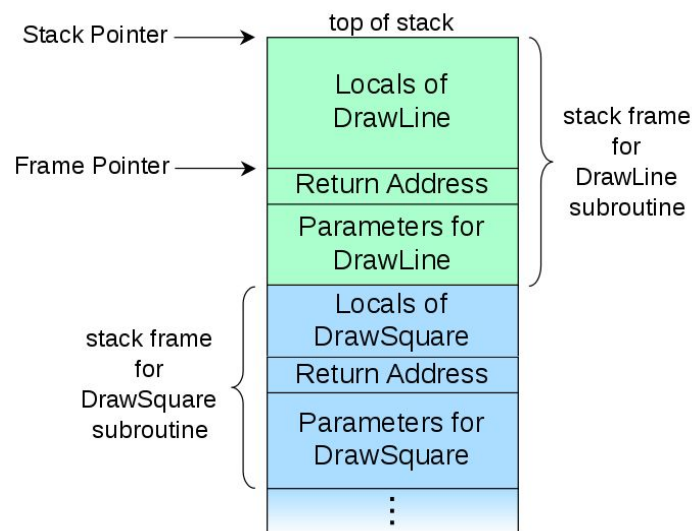


OS Lab 1

- Printf enhancement:
 - a. Add support for %o
 - Just add a case for '0', where base = 8 and goto number
 - b. Support %-, %+
 - Add case for '-' and make padc = ' ', right align in printnum()
 - I made this work by add two parameter to printnum:
 1. Int right_align: same as %- is set
 2. Int force_sign_sym: when %+ is set, the '+' or '-' symbol should put before the number
 - It's a bit hard to printing right align symbol under recursive call, instead, I make a new function doing this and calling the recursive function
- Stack backtrace, Implement mon_backtrace()
 - a. Read the stack information by writing inline assembly.
 - b. We need to read the %eip, %ebp, the first 5 argument and debug info
 - c. Debug info: the debuginfo_eip() function lack implementation of searching the source line number, there is a stab type N_SLINE representing that, by searching with `stab_binsearch(stabs, &lline, &rline, N_SLINE, addr) and put stabs[lline].n_desc into eip_line field, this work fine.
 - d. The calling stack layout is as follow:



So all above things is at fixed position from %esp, which can accessed writing little inline assembly.

- %eip: $[ebp + 4] - 4$
- N th parameter: $[ebp + 4 + i * 4]$

- Time command

The assembly command 'rdtsc' read the current clock, using to implement the mon_time() by running the command wrapped by rdtsc to get the clock value at start and end time.

```
__asm __volatile(  
    "rdtsc\n\t"  
    "movl %%edx, %0\n\t"  
    "movl %%eax, %1\n\t"  
    : "=r"(start_time_high), "=r"(start_time_low)  
    :  
    : "%eax", "%edx");
```

One thing to notice is %eax and %edx must make as Clobbers, as shows in picture last line.

