```c
absdiff.c

if (a < b)
    return b - a;
else
    return a - b;
```

```
absdiff:

    movl 8(%ebp), %eax

    movl 12(%ebp), %edx

    cmpl %edx, %eax

    ; update flags based on a - b

    ZF 0
    SF 1
    OF 0
```
\[ \text{jge } .L2 \]

\[ \rightarrow \text{jump does not happen here.} \]

\[ \text{movl } 12(\%ebp), \%eax } \]

\[ \text{\%eax } [2] \]

\[ \subl 8(\%ebp), \%eax \]

\[ \%eax [1] \]

\[ 2 - 1 = 1 \]

\[ \%eax \text{ has } b \; \text{ and } \%[\%ebp + 8] \text{ has } a. \]

\[ \therefore \text{ b - a is stored in } \%eax. \]

\[ \text{jmp } .L3 \]

\[ \text{\%L2} \]

\[ \text{movl } 8(\%ebp), \%eax \]

\[ \\text{\%L3} \]

\[ \text{popl } \%ebp \]

\[ \text{ret} \]

\[ \text{Condition for } \text{jge}: \]

\[ \text{SF} \land \text{OF} \]

\[ \text{\%L2} \]

\[ \text{Condition for } \text{jge}: \]

\[ \text{\n(SF} \land \text{OF)} \]

\[ \therefore \text{For our example,} \]

\[ \text{cond} = \text{\n(1} \land 0) \]

\[ = \text{\n1} = 0 \]

\[ (\therefore \text{only 1 bit}) \]

\[ \{ \text{\%eax [1]} \}

\[ \text{\therefore would be executed if } a > b. \]
To calculate the address of the function `absdiff`:

\[
\%eip = 0x08048413 \quad \text{(Address in PC)}
\]

Relative jump = \[0x\text{FF FF FF D9}\]

offset = 0x080483 EC

Actually, the `absdiff` function is at a smaller address than `main`.

\[0xFF FFFF D9\] is a negative number in 2's complement form.
Procedures

procedure call - 1 passing data
e.g. procedure parameters, return values.
2 transferring control.

Stack Frame

caller's frame

Frame pointer
%ebp

Stack pointer
%esp

Registers
Local variables
Arguments
Return address

Saved %ebp
Registers
Local variables
Args
Return address

current frame

added if current fn calls another fn.