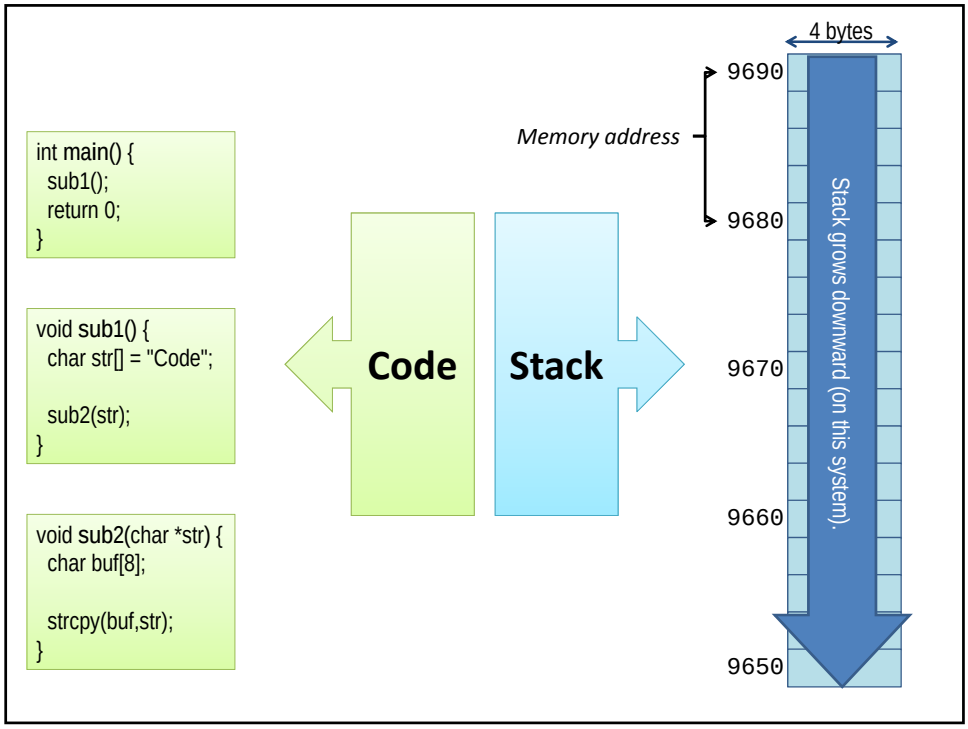
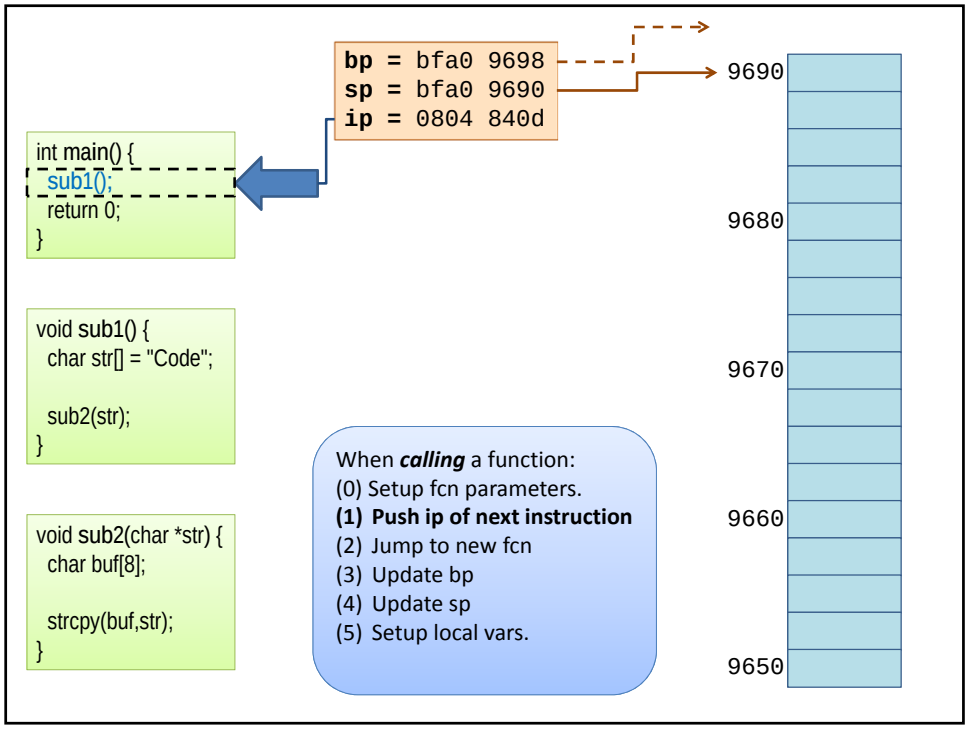
Buffer Overflow Example

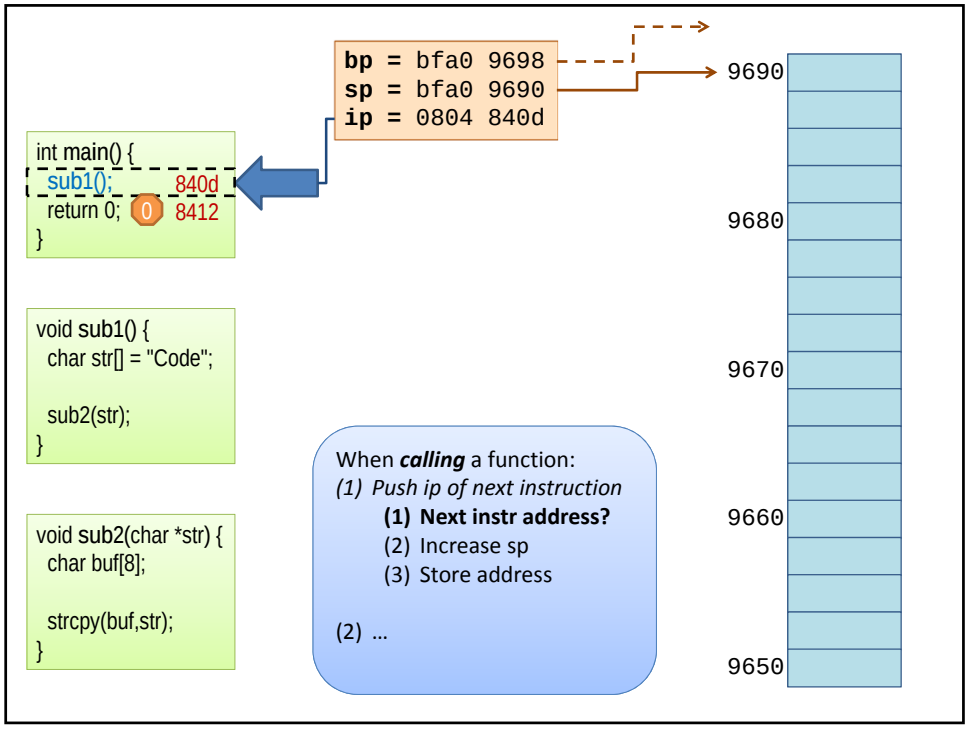
Slides done by Magnus Almgren

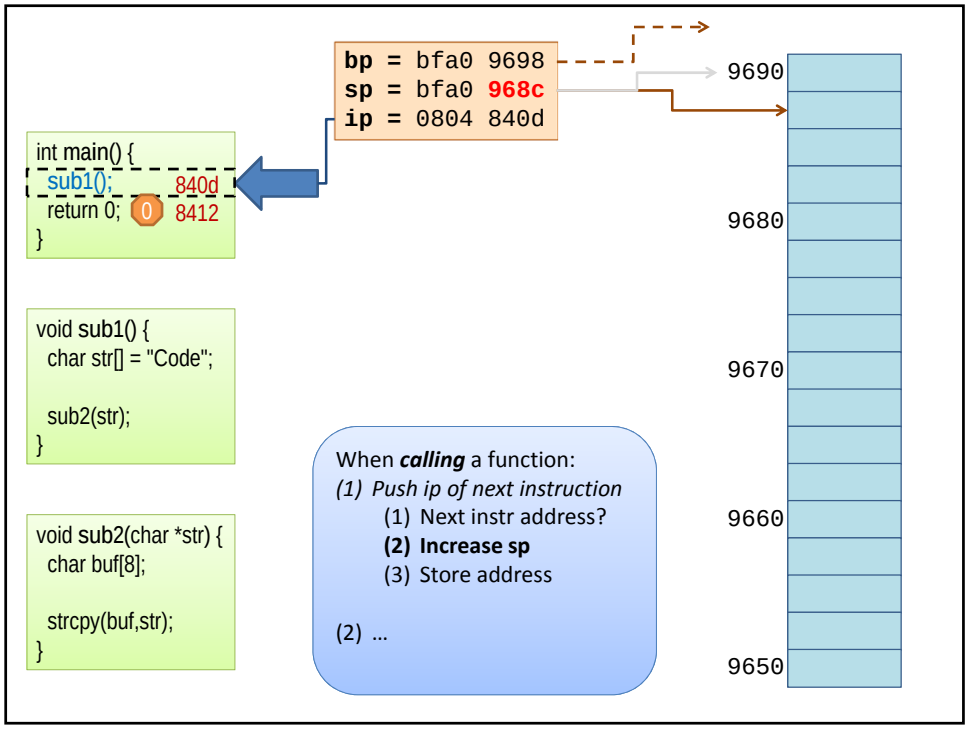
Source code of program example

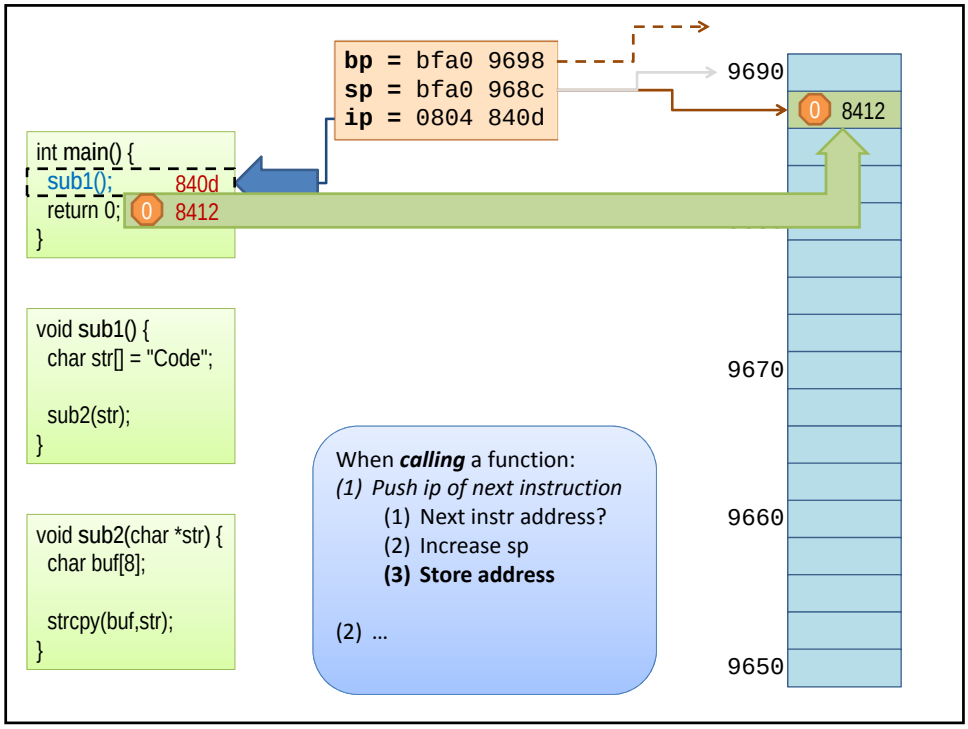
```
#include <string.h>

void sub2(char *str) {
  char buf[8];

  strcpy(buf,str);
}

void sub1() {
  char str[] = "Code";

  sub2(str);
}

int main() {
  sub1();
  return 0;
}
```

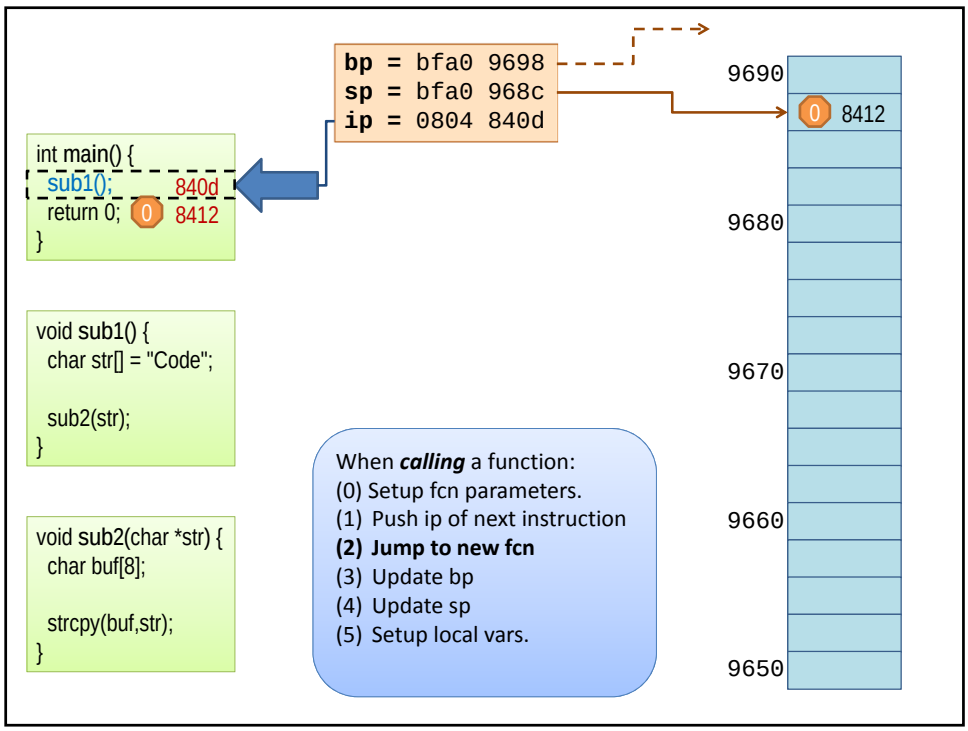

4 bytes
Memory address
9690
9680
9670
9660
9650
Stack grows downward (on this system).
Code
Stack
int main() {
sub1();
return 0;
}
void sub1() {
char str[] = "Code";
sub2(str);
}
void sub2(char *str) {
char buf[8];
strcpy(buf,str);
}

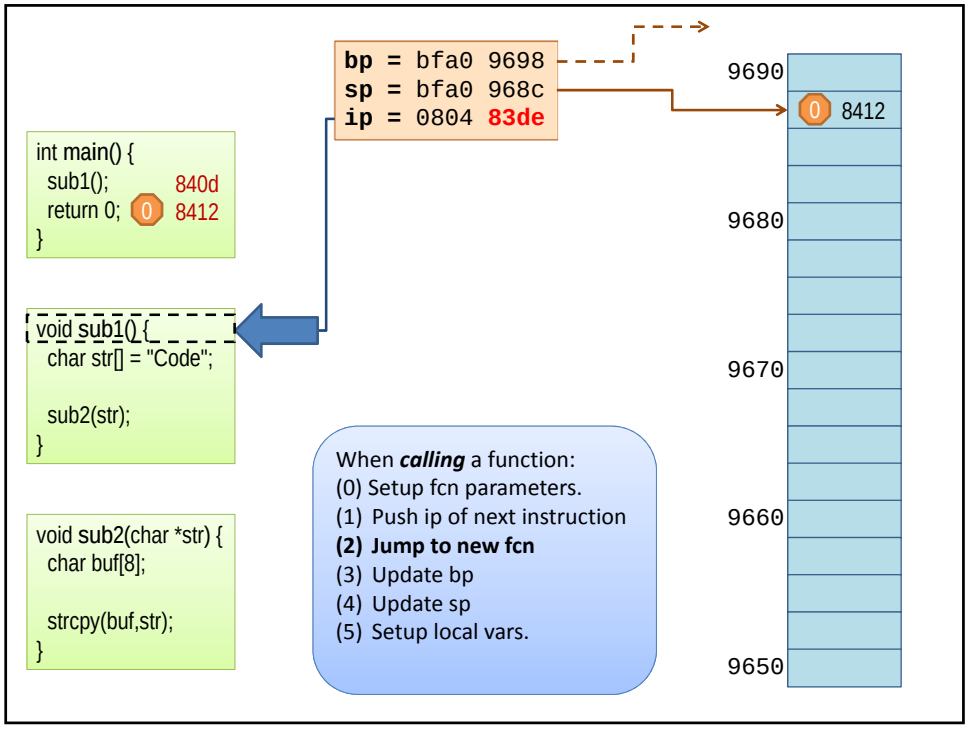

bp = bfa0 9698
sp = bfa0 9690
ip = 0804 840d
9690
9680
9670
9660
9650
int main() {
sub1();
return 0;
}
void sub1() {
char str[] = "Code";
sub2(str);
}
void sub2(char *str) {
char buf[8];
strcpy(buf,str);
}
When calling a function:
(0) Setup fcn parameters.
(1) Push ip of next instruction
(2) Jump to new fcn
(3) Update bp
(4) Update sp
(5) Setup local vars.

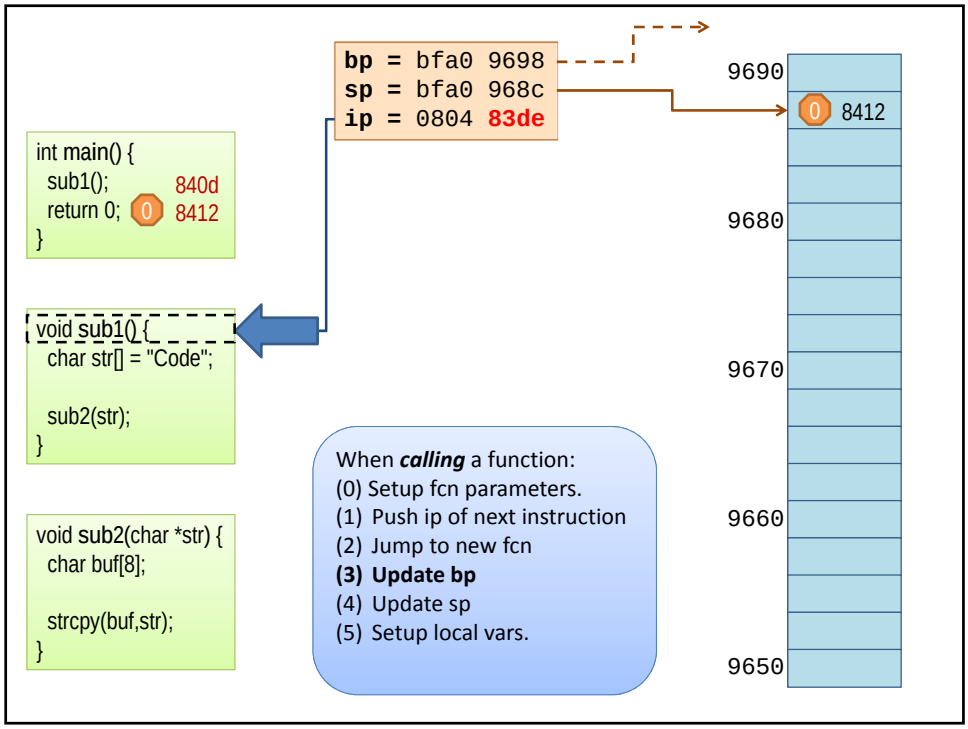

bp = bfa0 9698
sp = bfa0 9690
ip = 0804 840d
int main() {
sub1();
840d
return 0;
0
8412
}
void sub1() {
char str[] = "Code";
sub2(str);
}
void sub2(char *str) {
char buf[8];
strcpy(buf,str);
}
When calling a function:
(1) Push ip of next instruction
(1) Next instr address?
(2) Increase sp
(3) Store address
(2) …
9690
9680
9670
9660
9650

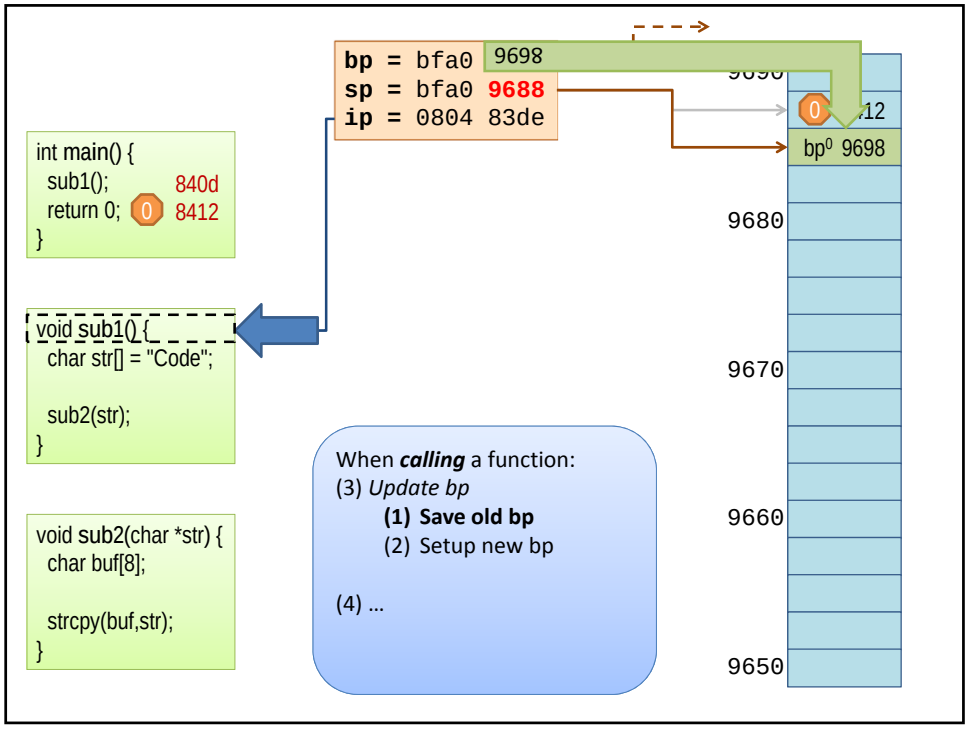

bp = bfa0 9698
sp = bfa0 968c
ip = 0804 840d
int main() {
sub1();
840d
return 0;
0
8412
}
void sub1() {
char str[] = "Code";
sub2(str);
}
void sub2(char *str) {
char buf[8];
strcpy(buf,str);
}
When calling a function:
(1) Push ip of next instruction
(1) Next instr address?
(2) Increase sp
(3) Store address
(2) …
9690
9680
9670
9660
9650

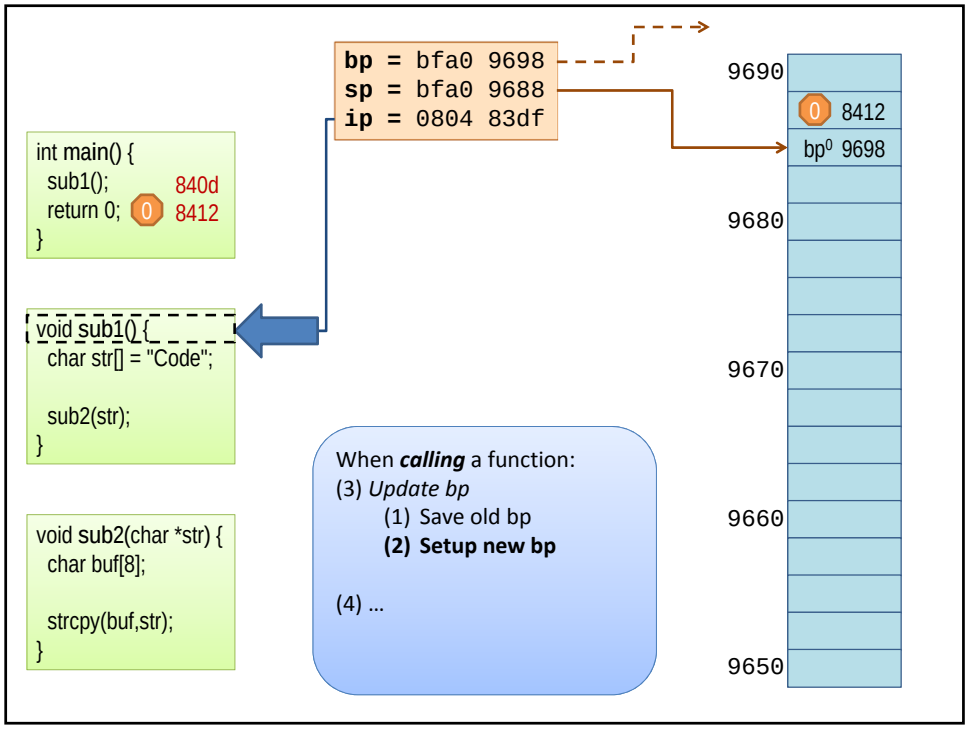

bp = bfa0 9698
sp = bfa0 968c
ip = 0804 840d
9690
0
8412
int main() {
sub1();
840d
return 0;
0
8412
}
void sub1() {
char str[] = "Code";
sub2(str);
}
9670
9660
void sub2(char *str) {
char buf[8];
strcpy(buf,str);
}
When calling a function:
(1) Push ip of next instruction
(1) Next instr address?
(2) Increase sp
(3) Store address
(2) ...
9650

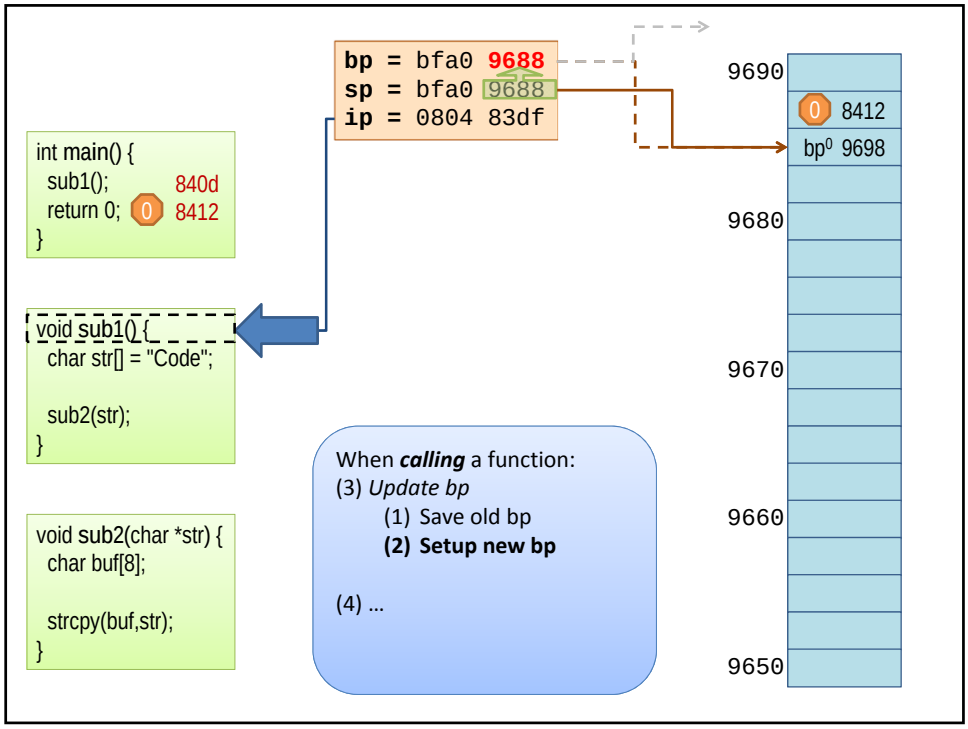

bp = bfa0 9698
sp = bfa0 968c
ip = 0804 840d
9690
0
8412
int main() {
sub1();
840d
return 0;
0
8412
}
9680
void sub1() {
char str[] = "Code";
sub2(str);
}
9670
9660
void sub2(char *str) {
char buf[8];
strcpy(buf,str);
}
When calling a function:
(0) Setup fcn parameters.
(1) Push ip of next instruction
(2) Jump to new fcn
(3) Update bp
(4) Update sp
(5) Setup local vars.
9650

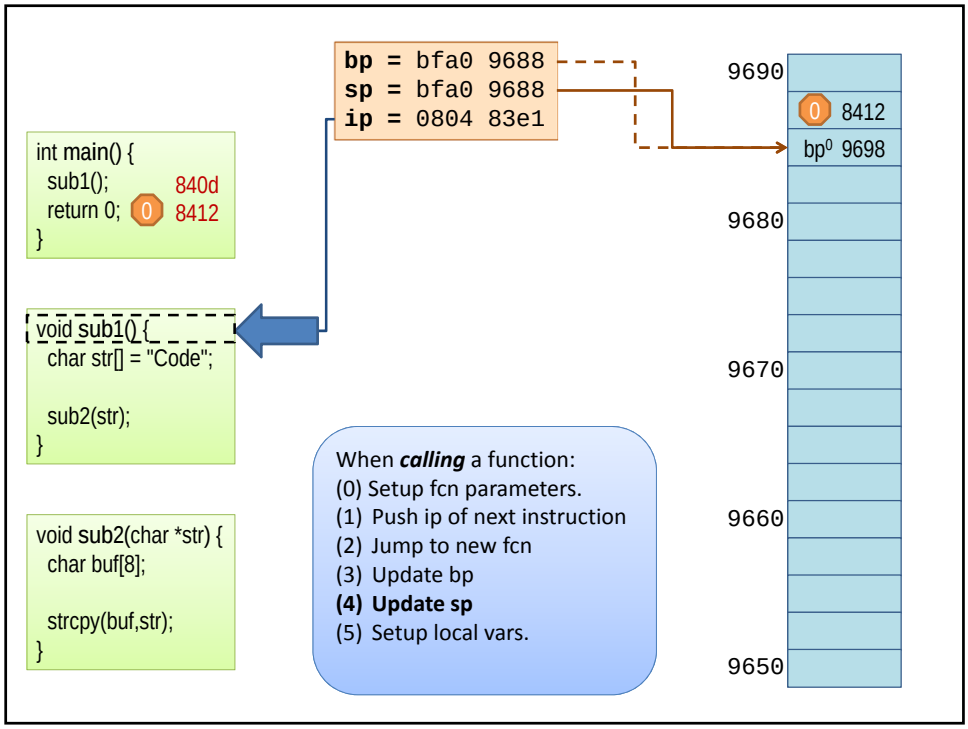

bp = bfa0 9698
sp = bfa0 968c
ip = 0804 83de
int main() {
sub1();
840d
return 0;
0
8412
}
void sub1() {
char str[] = "Code";
sub2(str);
}
void sub2(char *str) {
char buf[8];
strcpy(buf,str);
}
When calling a function:
(0) Setup fcn parameters.
(1) Push ip of next instruction
(2) Jump to new fcn
(3) Update bp
(4) Update sp
(5) Setup local vars.
9690
0
8412
9680
9670
9660
9650

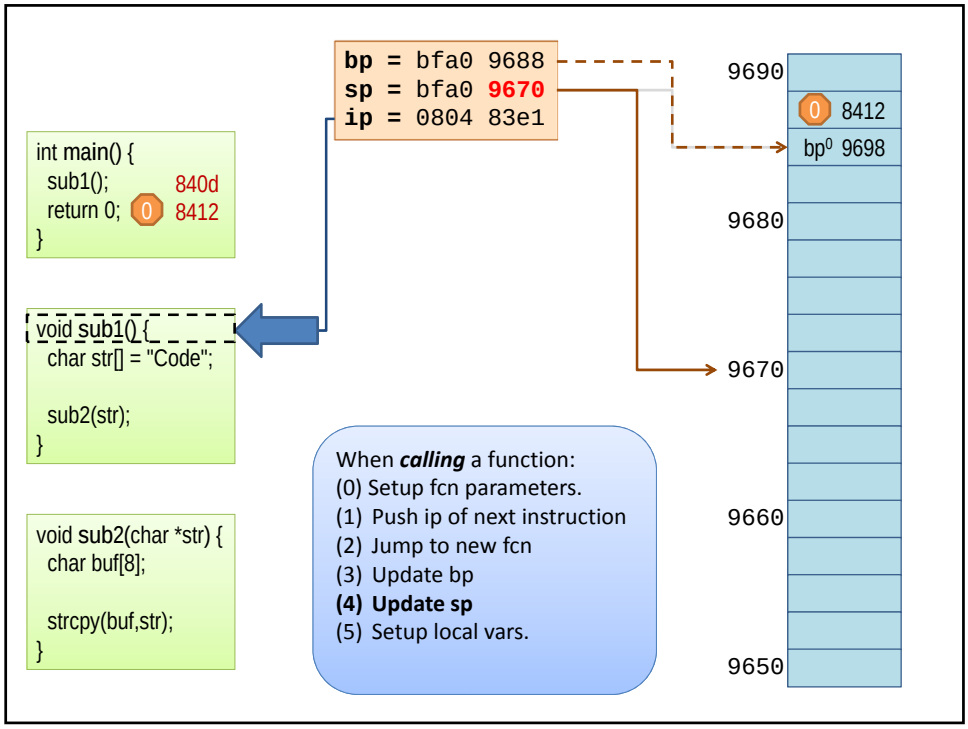

bp = bfa0 9698
sp = bfa0 968c
ip = 0804 83de
int main() {
sub1();
840d
return 0;
0
8412
}
void sub1() {
char str[] = "Code";
sub2(str);
}
void sub2(char *str) {
char buf[8];
strcpy(buf,str);
}
When calling a function:
(0) Setup fcn parameters.
(1) Push ip of next instruction
(2) Jump to new fcn
(3) Update bp
(4) Update sp
(5) Setup local vars.
9690
0
8412
9680
9670
9660
9650

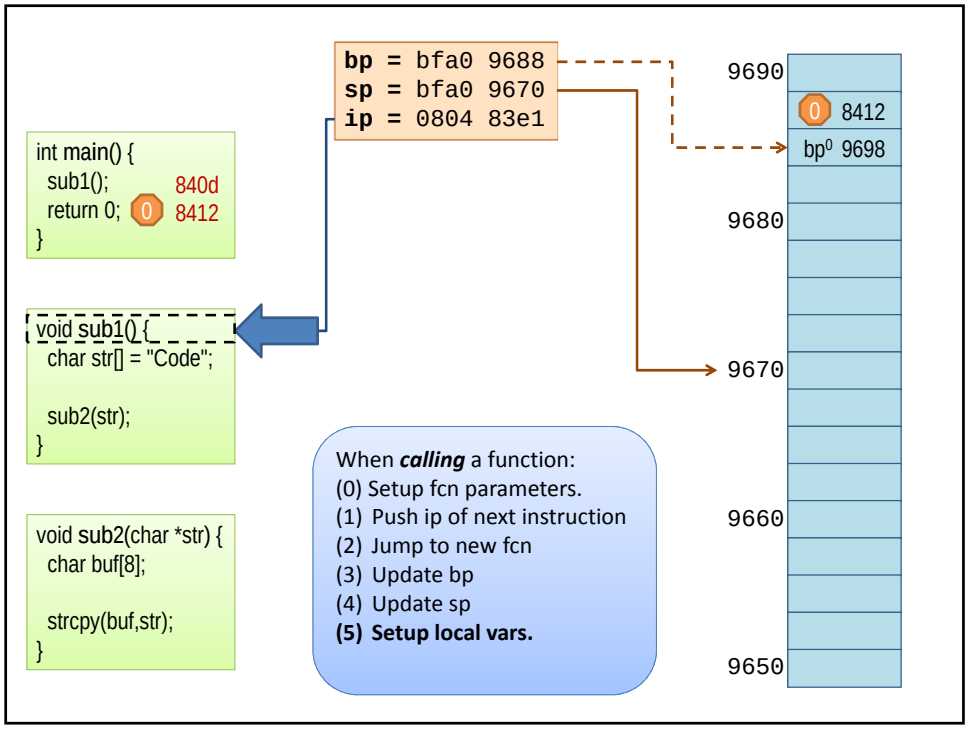

bp = bfa0 9698
sp = bfa0 9688
ip = 0804 83de
int main() {
sub1();
return 0;
}
840d
8412
void sub1() {
char str[] = "Code";
sub2(str);
}
void sub2(char *str) {
char buf[8];
strcpy(buf,str);
}
When calling a function:
(3) Update bp
(1) Save old bp
(2) Setup new bp
(4) ...
bp0 9698
9680
9670
9660
9650

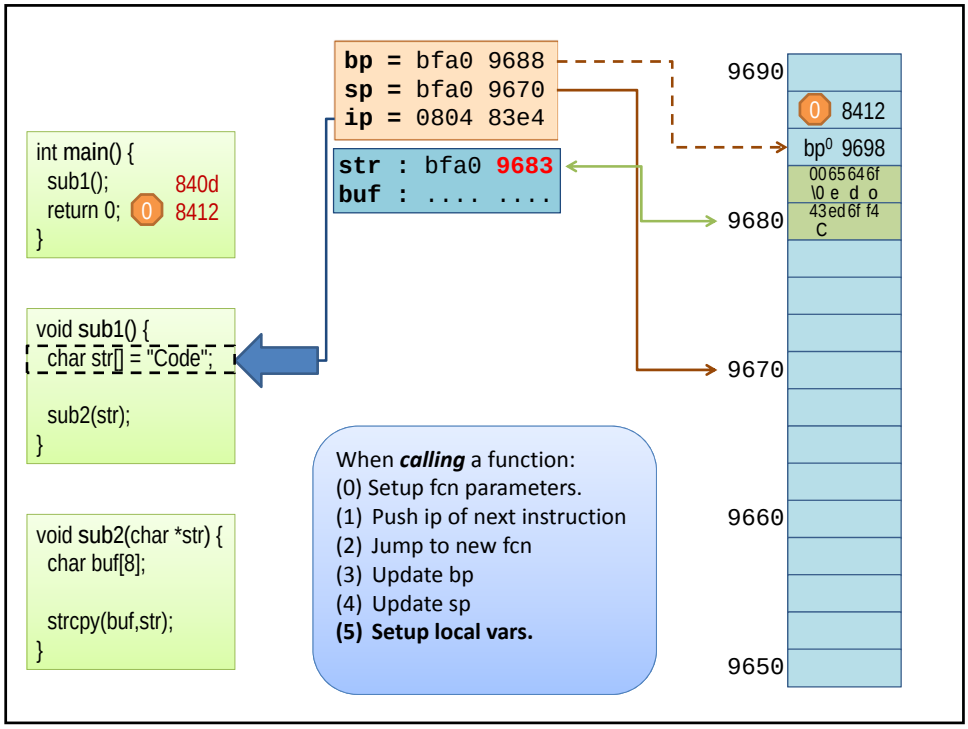

bp = bfa0 9698
sp = bfa0 9688
ip = 0804 83df
int main() {
sub1();
return 0;
}
840d
8412
void sub1() {
char str[] = "Code";
sub2(str);
}
void sub2(char *str) {
char buf[8];
strcpy(buf,str);
}
When calling a function:
(3) Update bp
(1) Save old bp
(2) Setup new bp
(4) ...
8412
bp0 9698
9690
9680
9670
9660
9650

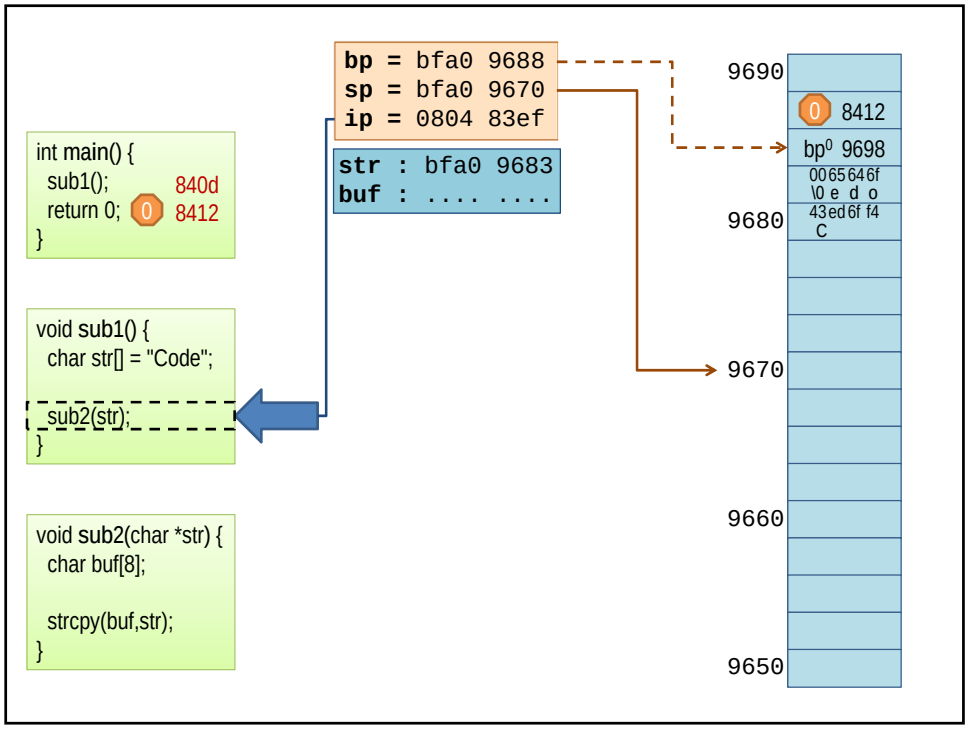

bp = bfa0 9688
sp = bfa0 9688
ip = 0804 83df
int main() {
sub1(); 840d
return 0; 0 8412
}
void sub1() {
char str[] = "Code";
sub2(str);
}
void sub2(char *str) {
char buf[8];
strcpy(buf,str);
}
When calling a function:
(3) Update bp
(1) Save old bp
(2) Setup new bp
(4) ...
9690
0 8412
bp0 9698
9680
9670
9660
9650

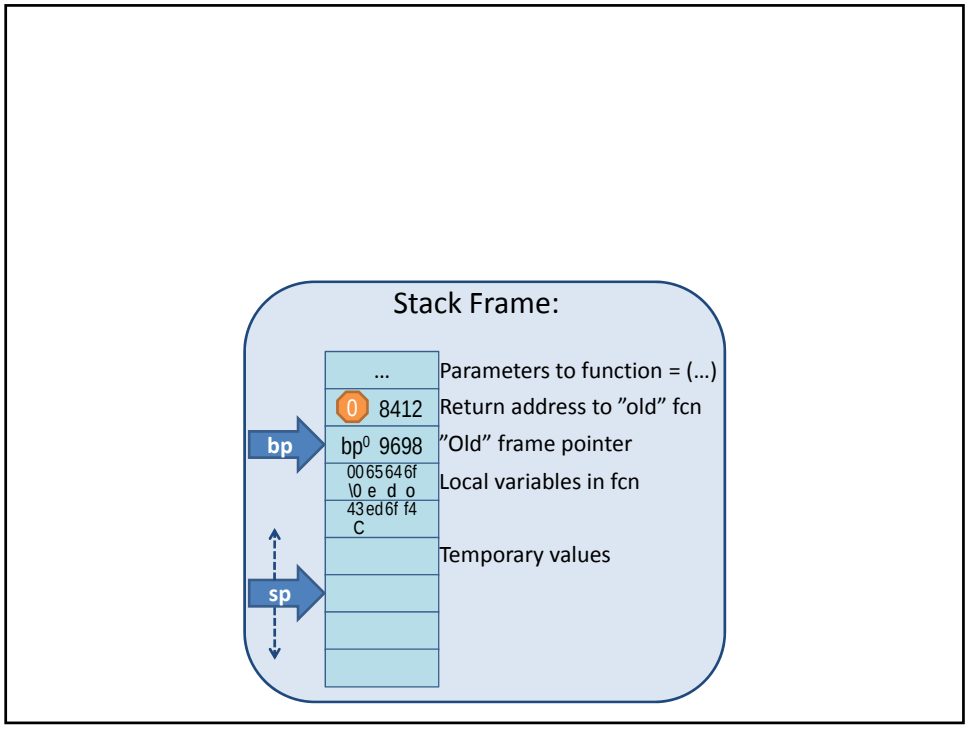

bp = bfa0 9688
sp = bfa0 9688
ip = 0804 83e1
int main() {
sub1(); 840d
return 0; 0 8412
}
void sub1() {
char str[] = "Code";
sub2(str);
}
void sub2(char *str) {
char buf[8];
strcpy(buf,str);
}
When calling a function:
(0) Setup fcn parameters.
(1) Push ip of next instruction
(2) Jump to new fcn
(3) Update bp
(4) Update sp
(5) Setup local vars.
9690
0 8412
bp0 9698
9680
9670
9660
9650

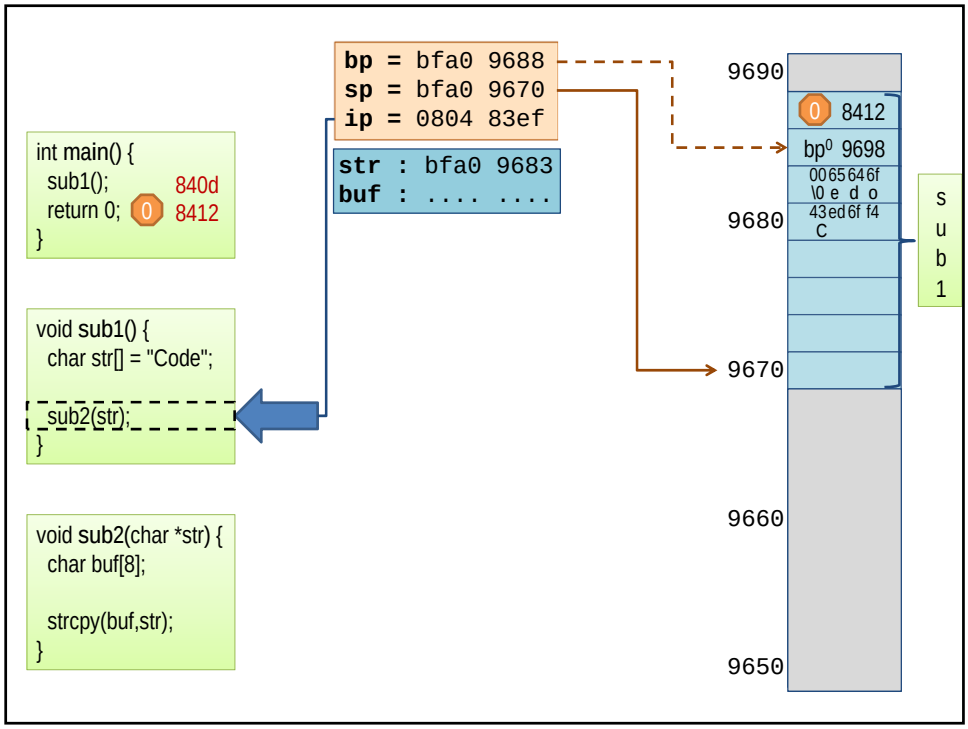

bp = bfa0 9688
sp = bfa0 9670
ip = 0804 83e1
int main() {
sub1();
return 0;
}
840d
8412
void sub1() {
char str[] = "Code";
sub2(str);
}
void sub2(char *str) {
char buf[8];
strcpy(buf,str);
}
When calling a function:
(0) Setup fcn parameters.
(1) Push ip of next instruction
(2) Jump to new fcn
(3) Update bp
(4) Update sp
(5) Setup local vars.
9690
8412
bp0 9698
9680
9670
9660
9650

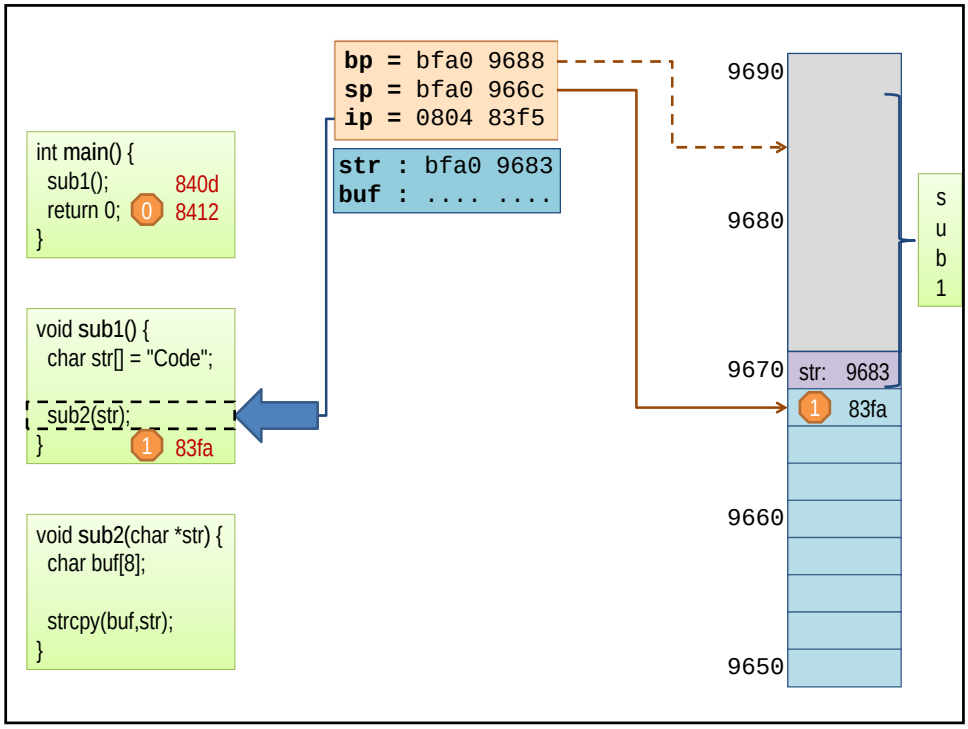

bp = bfa0 9688
sp = bfa0 9670
ip = 0804 83e1
int main() {
sub1();
return 0;
}
840d
8412
void sub1() {
char str[] = "Code";
sub2(str);
}
void sub2(char *str) {
char buf[8];
strcpy(buf,str);
}
When calling a function:
(0) Setup fcn parameters.
(1) Push ip of next instruction
(2) Jump to new fcn
(3) Update bp
(4) Update sp
(5) Setup local vars.
9690
8412
bp0 9698
9680
9670
9660
9650

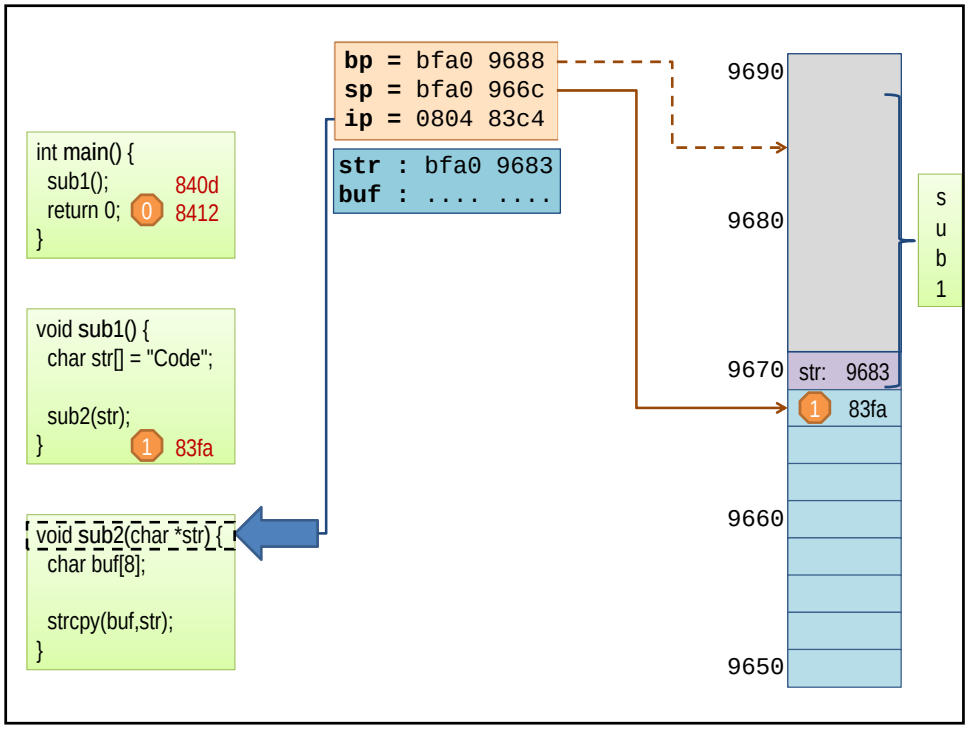

bp = bfa0 9688
sp = bfa0 9670
ip = 0804 83e4
str : bfa0 9683
buf : ....
int main() {
sub1();
return 0;
}
840d
8412
void sub1() {
char str[] = "Code";
sub2(str);
}
void sub2(char *str) {
char buf[8];
strcpy(buf,str);
}
When calling a function:
(0) Setup fcn parameters.
(1) Push ip of next instruction
(2) Jump to new fcn
(3) Update bp
(4) Update sp
(5) Setup local vars.
9690
8412
bp0 9698
00 65 64 6f
\0 e d o
43 ed 6f f4
C
9680
9670
9660
9650

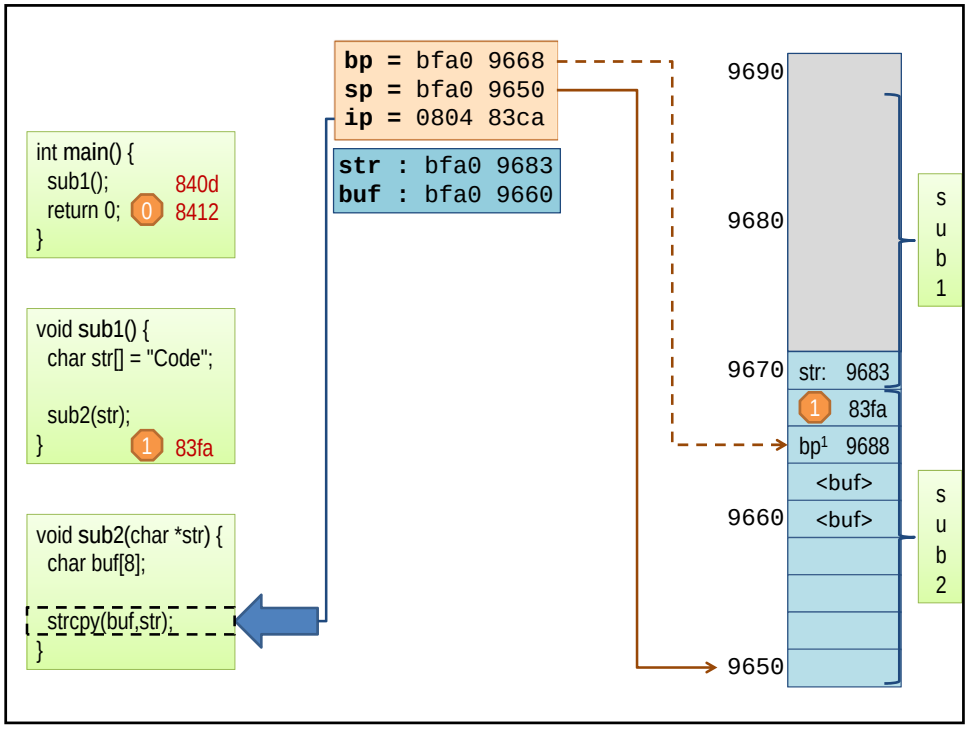

bp = bfa0 9688
sp = bfa0 9670
ip = 0804 83ef
str : bfa0 9683
buf : ....
int main() {
sub1();
return 0;
}
840d
8412
void sub1() {
char str[] = "Code";
sub2(str);
}
void sub2(char *str) {
char buf[8];
strcpy(buf,str);
}
9690
8412
bp0 9698
00 65 64 6f
\0 e d o
43 ed 6f f4
C
9680
9670
9660
9650

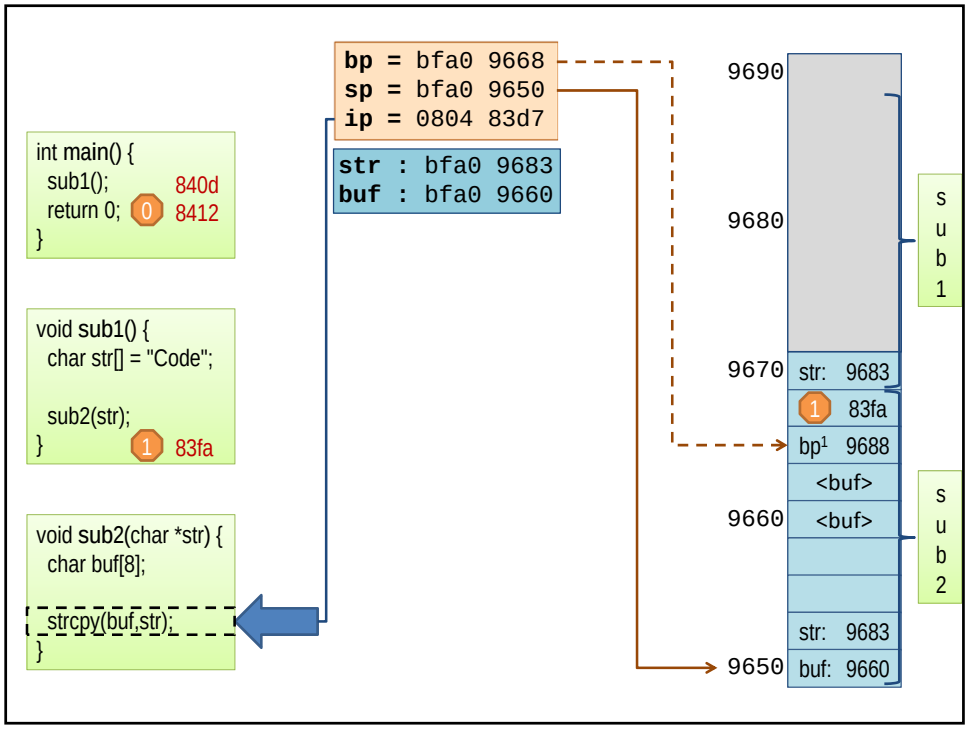

Stack Frame:
...
Parameters to function = (...)
0 8412
Return address to "old" fcn
bp
bp0 9698
"Old" frame pointer
00 65 64 6f
\0 e d o
Local variables in fcn
43 ed 6f f4
C
Temporary values
sp

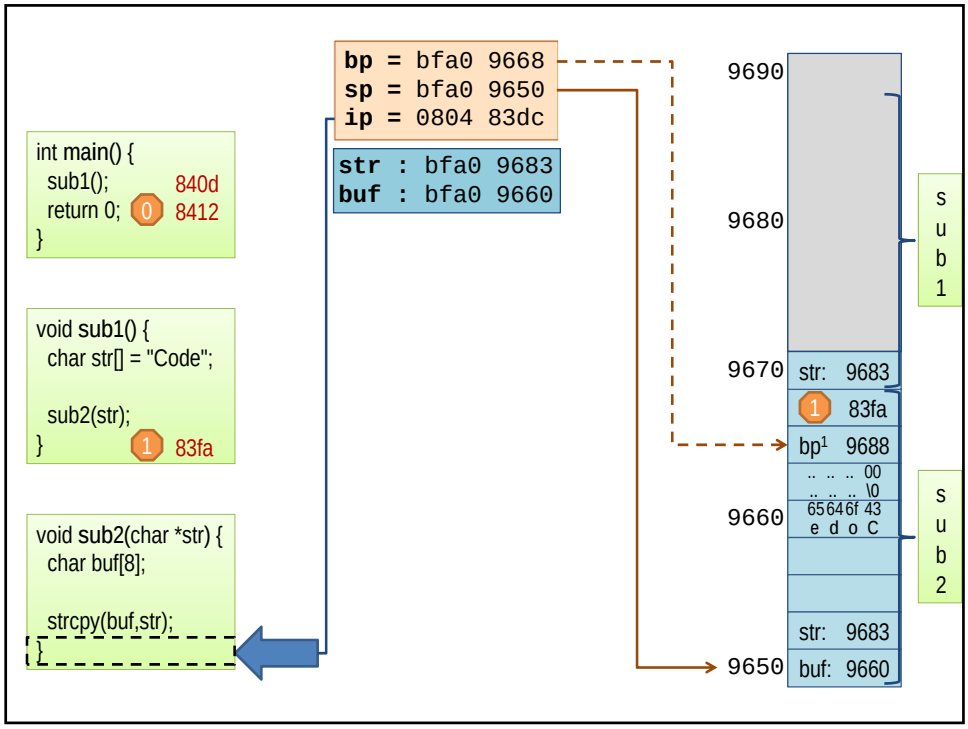

bp = bfa0 9688
sp = bfa0 9670
ip = 0804 83ef
str : bfa0 9683
buf : ....
int main() {
sub1();
return 0;
840d
8412
}
void sub1() {
char str[] = "Code";
sub2(str);
}
void sub2(char *str) {
char buf[8];
strcpy(buf,str);
}
9690
9680
9670
9660
9650
0 8412
bp0 9698
00 65 64 6f
\0 e d o
43 ed 6f f4
C
sub1


bp = bfa0 9688
sp = bfa0 966c
ip = 0804 83f5
str : bfa0 9683
buf : ....
int main() {
sub1(); 840d
return 0; 0 8412
}
void sub1() {
char str[] = "Code";
sub2(str);
} 1 83fa
void sub2(char *str) {
char buf[8];
strcpy(buf,str);
}
9690
9680
9670
9660
9650
str: 9683
1 83fa
sub1


bp = bfa0 9688
sp = bfa0 966c
ip = 0804 83c4
str : bfa0 9683
buf : ....
int main() {
sub1(); 840d
return 0; 0 8412
}
void sub1() {
char str[] = "Code";
sub2(str);
} 1 83fa
void sub2(char *str) {
char buf[8];
strcpy(buf,str);
}
9690
9680
9670
9660
9650
str: 9683
1 83fa
sub1

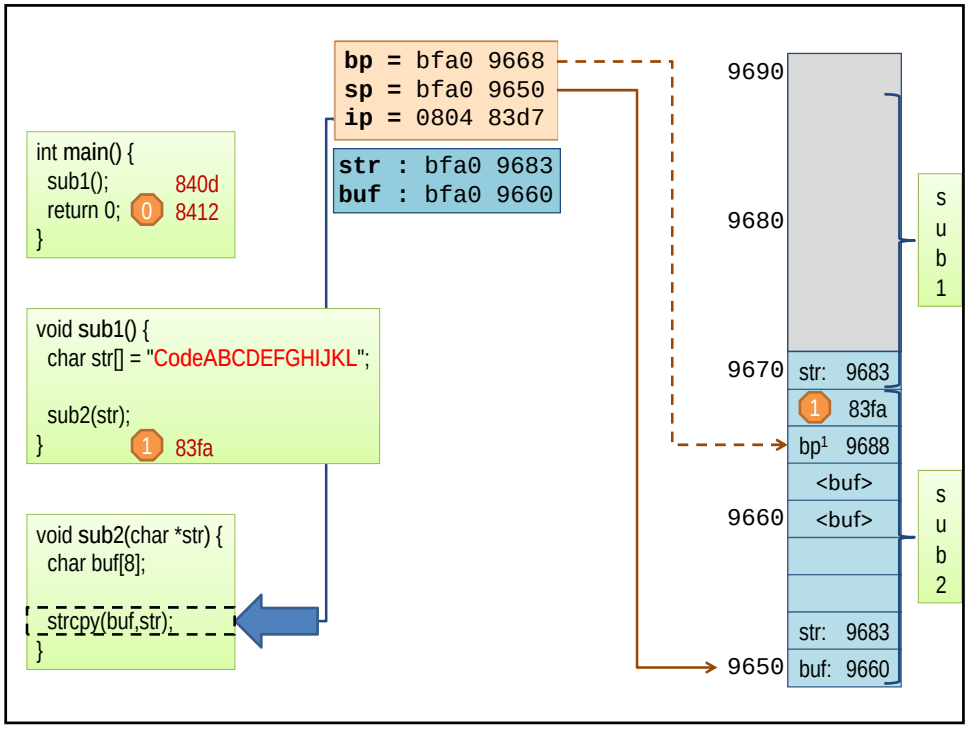

bp = bfa0 9668
sp = bfa0 9650
ip = 0804 83ca
str : bfa0 9683
buf : bfa0 9660
int main() {
sub1();
840d
return 0;
0
8412
}
void sub1() {
char str[] = "Code";
sub2(str);
}
1
83fa
void sub2(char *str) {
char buf[8];
strcpy(buf,str);
}
9690
9680
9670
9660
9650
str: 9683
1
83fa
bp1 9688
<buf>
<buf>
sub1
sub2

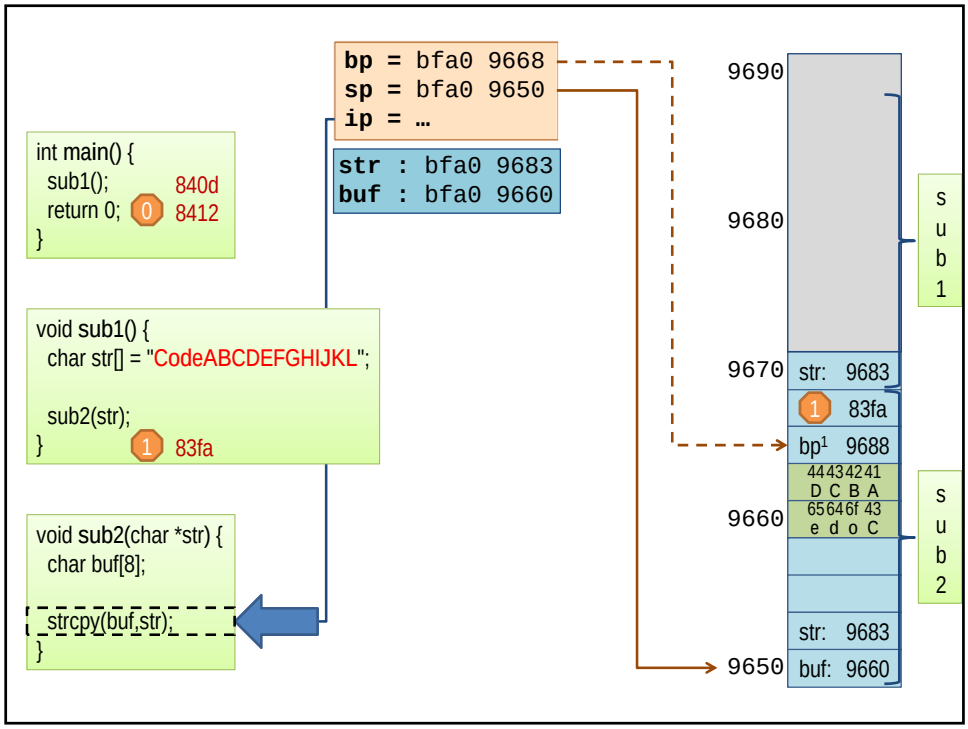

bp = bfa0 9668
sp = bfa0 9650
ip = 0804 83d7
str : bfa0 9683
buf : bfa0 9660
int main() {
sub1();
840d
return 0;
0
8412
}
void sub1() {
char str[] = "Code";
sub2(str);
}
1
83fa
void sub2(char *str) {
char buf[8];
strcpy(buf,str);
}
9690
9680
9670
9660
9650
str: 9683
1
83fa
bp1 9688
<buf>
<buf>
str: 9683
buf: 9660
sub1
sub2

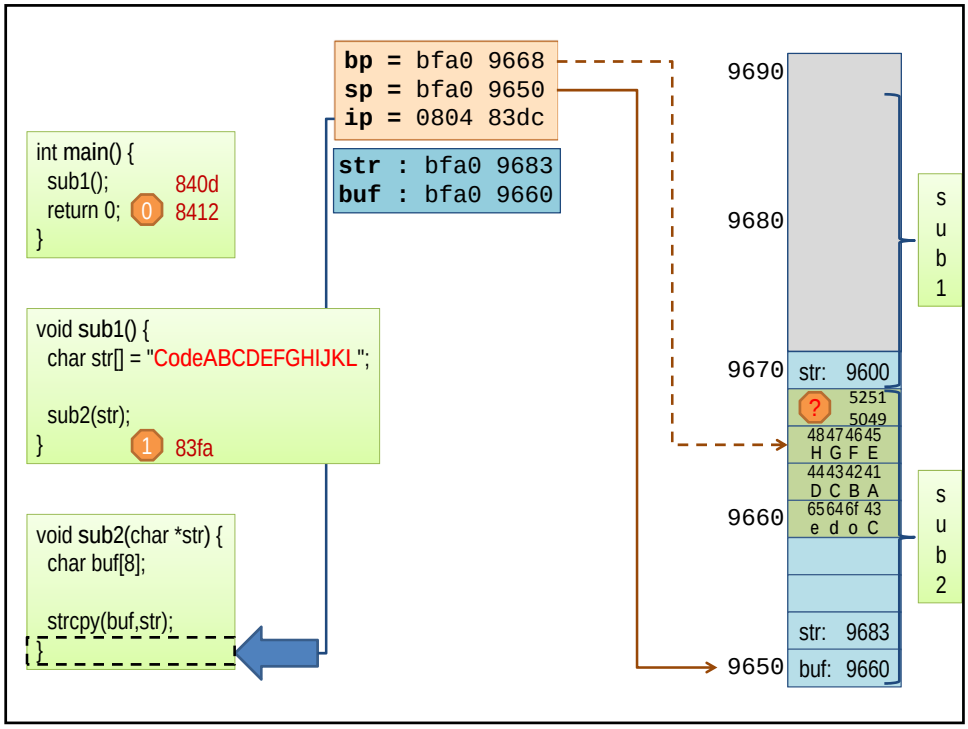


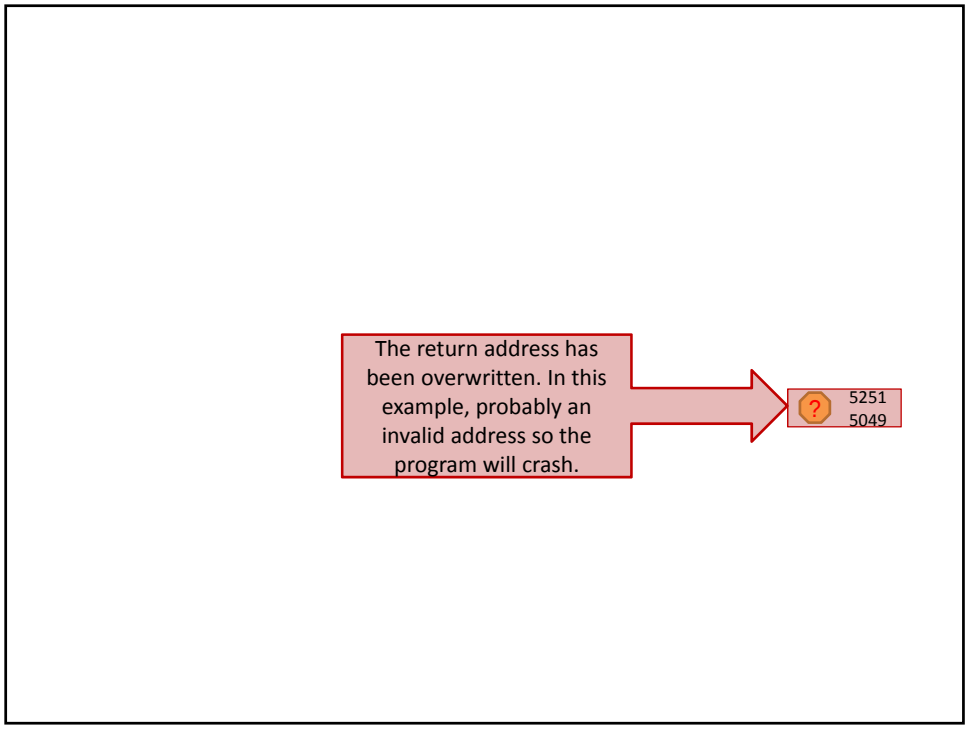
What if the string **str** was longer than 5 characters?
(4 characters + ending '\0'-character)

Let's back up a few steps …


bp = bfa0 9668
sp = bfa0 9650
ip = 0804 83d7
str : bfa0 9683
buf : bfa0 9660
int main() {
sub1(); 840d
return 0; 0 8412
}
void sub1() {
char str[] = "CodeABCDEFGHIJKL";
sub2(str);
} 1 83fa
void sub2(char *str) {
char buf[8];
strcpy(buf,str);
}
9690
9680
9670
9660
9650
str: 9683
1 83fa
bp1 9688
<buf>
<buf>
str: 9683
buf: 9660
sub1
sub2


bp = bfa0 9668
sp = bfa0 9650
ip = …
str : bfa0 9683
buf : bfa0 9660
int main() {
sub1(); 840d
return 0; 0 8412
}
void sub1() {
char str[] = "CodeABCDEFGHIJKL";
sub2(str);
} 1 83fa
void sub2(char *str) {
char buf[8];
strcpy(buf,str);
}
9690
9680
9670
9660
9650
str: 9683
1 83fa
bp1 9688
44434241
D C B A
65646f43
e d o C
str: 9683
buf: 9660
sub1
sub2


bp = bfa0 9668
sp = bfa0 9650
ip = 0804 83dc
str : bfa0 9683
buf : bfa0 9660
int main() {
sub1(); 840d
return 0; 0 8412
}
void sub1() {
char str[] = "CodeABCDEFGHIJKL";
sub2(str);
} 1 83fa
void sub2(char *str) {
char buf[8];
strcpy(buf,str);
}
9690
9680
9670
9660
9650
str: 9600
5251
5049
48474645
H G F E
44434241
D C B A
65646f43
e d o C
str: 9683
buf: 9660
sub1
sub2


The return address has been overwritten. In this example, probably an invalid address so the program will crash.
5251
5049

Trick: Jump back ***into*** your buffer