Function Pointer

Take this C code as example:

```
// this is defined globally.
void fun1()
{
}
// These lines are part of some function
void (*fptr)() = fun1;
fptr();
```

Generated assembly code:

```
movl $fun1, 12(%esp)
movl 12(%esp), %eax
call  *%eax
```

Location of local variables of the stack (local variables are explained here (memorymanagement.html))

```
fptr => 12(%esp)
```

The use of registers as temporary memory is described here (arithmeticop.html#tempVaribaleUsage)
Comments on generated assembly code:

```
# fptr = fun1
   movl    $fun1, 12(%esp)

# tmp = fptr
   movl    12(%esp), %eax

# call the function using pointer i.e. tmp()
   call    *%eax
```

NOTE: Here register esp is being used to refer to the local variables instead of ebp. This is up to the compiler to either use esp or ebp to index into the stack.

‹ Array Indexing (/cin/array.html)

up (/cin/cin.html)