

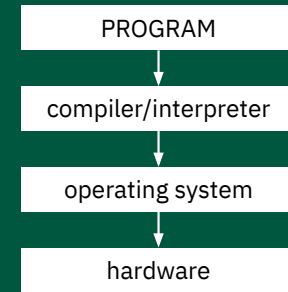
CMSC 124

Design and Implementation
of Programming Languages

Kristine Bernadette Pelaez
Institute of Computer Science
University of the Philippines Los Baños



Language Implementation Methods



Implementation Methods

1. Compilation
2. Interpretation
3. Hybrid Systems

1. Compilation

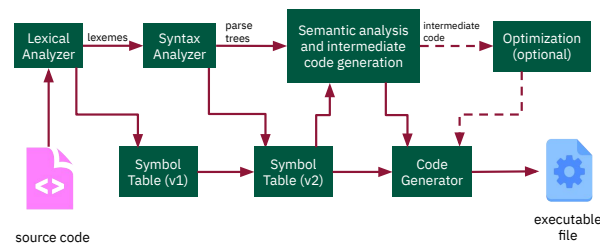
1. Compilation

programs are translated
to *machine code*

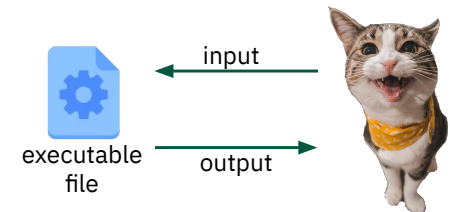
1. Compilation

Examples:
C, C++, COBOL

1. Compilation



1. Compilation



1. Compilation

fastest among the methods
of *implementation*

1. Compilation

the *von Neumann bottleneck*
limits the speed of programs

2. Interpretation

2. Interpretation

programs are interpreted
by another program
called *an interpreter*

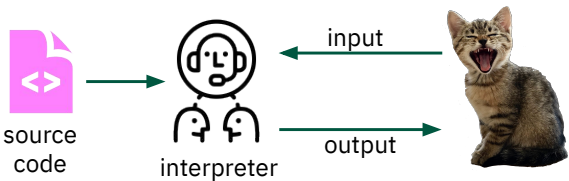
2. Interpretation

lexical, syntax, & semantic
analysis are still done but
no translation to machine code happens

2. Interpretation

statements are
decoded and executed
one by one

2. Interpretation



2. Interpretation

Examples:
Python, PHP, JavaScript

2. Interpretation

this method is
easier to implement

2. Interpretation

allows *easier source-level debugging* since statements are interpreted one by one

2. Interpretation

however, it is *10 to 100 times slower*

2. Interpretation

```
for subj in courses:
    print("Grade in",subj)
    input("Grade:")
```

2. Interpretation

speed is affected by *how fast the interpreter can decode a statement*

3. Hybrid Implementation Systems

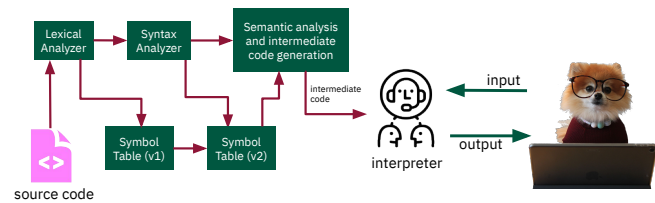
3. Hybrid Implementation Systems

a *compromise* between compilers and interpreters

3. Hybrid Implementation Systems

source code is *translated* to an *intermediate language* that is easier to *interpret*

3. Hybrid Implementation Systems



3. Hybrid Implementation Systems

faster than interpretation since statements are *decoded only once*

3. Hybrid Implementation Systems

Examples:

Perl, Java

Preprocessors

Preprocessors

accepts a *program as input* and returns the *modified version as output*

KBPNalaez, ICS, UPB, 2020.

28

KBPNalaez, ICS, UPB, 2020.

30

Preprocessors

a program that
processes a program before it is compiled

Preprocessors

```
#include <stdio.h>
#ifdef QUEUE_H
#define TRUE 1
#endif
```

KBPNalaez, ICS, UPB, 2020.

31

KBPNalaez, ICS, UPB, 2020.

CMSC 124

*Design and Implementation
of Programming Languages*

Kristine Bernadette Pelaez
Institute of Computer Science
University of the Philippines Los Baños



32