Lecture 0: Systems Software

Pawel Wocjan

Course Organization

Syllabus

http://www.cs.ucf.edu/~wocjan/Teaching/2016_Fall/SS

Systems Software

Course Outline

This course is designed to provide a fundamental understanding of real and virtual machines as a language processor. We will study the processor as an instruction interpreter. Compilers, assemblers, and virtual machines will be presented as systems software for program development. An introduction to operating systems will be given.

Course Topics

Introduction to compilers and interpreters, virtual machines, computer architecture and assembler, loaders and linkers, macro-processors, run time environments and operating systems

Prerequisites

COP 3502 – Computer Science I, COP 3402 Intro to Programming with C, CDA 3101 Computer Organization

What is Systems Software?

Systems Software consists of a set of programs that

- support the operation of a computer system,
- help the programmer simplify the programming process, and
- create an environment to run application software efficiently.

Examples of Systems Software are:

- **Text Editor**
- **Compiler**
- **Loader**
- **Linker**
- **Debugger**
- **Assembler**
- **Operating System**

Classification of Systems Software in Two Groups

Software to create a Program Development Environment

- Text Editor
- Compiler
- Assembler
- Linker
- Debugger (low-level)

Software to create a Run-Time Environment

- Operating System
- Loader
- Dynamic Linker
- Program Libraries

Program Development Environment

Text Editor

Permits the creation and editing of text files

Compiler

Translates programs written in a high level language to machine code (Assembly Language)

Assembler

Translates programs written in assembly language to object code (binary)

Static Linker

Combines and resolves references between object programs and creates the executable code

Debugger

Is used to debug executable programs and their related object code and source program

Run-Time Environment

Loader

Loads an executable code and starts its execution

Libraries

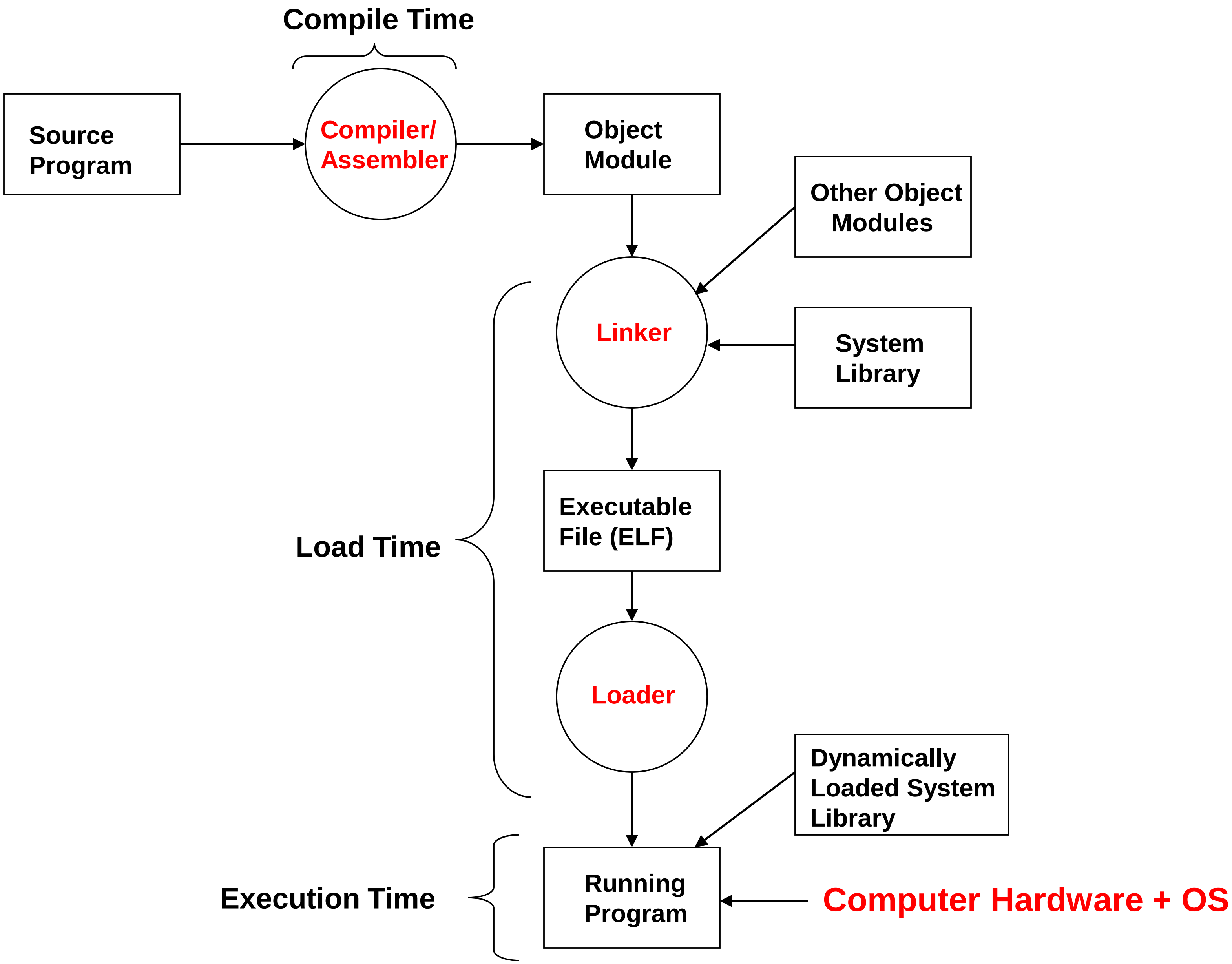
Precompiled programs that create a set of functions for use by other programs

Dynamic Linker

Loads and links shared libraries at run-time

Operating System

- Is an event driven program that provides an abstraction of the computer system
- handles all resources efficiently, creates an environment for application programs to run
- Provides a friendly interface between the user and the computer system



Compile Time

Source Program

Compiler/ Assembler

Object Module

Other Object Modules

Linker

System Library

Load Time

Executable File (ELF)

Loader

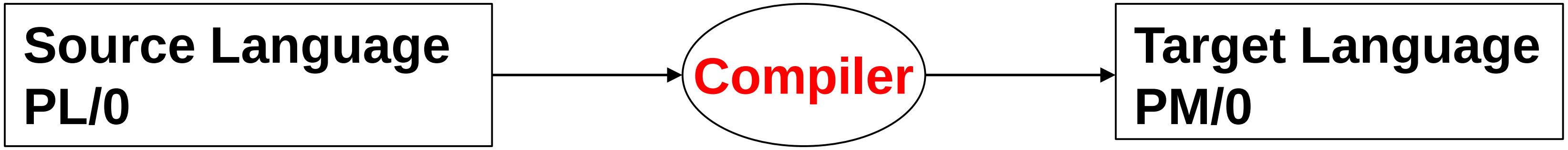
Dynamically Loaded System Library

Execution Time

Running Program

Computer Hardware + OS

Programming Project



- **PL/0** is an instructional language that is much simpler than the general purpose language Pascal. It was originally introduced by Nikolas Wirth in the 70s.

- **PM/0** is the instruction set architecture for a simple virtual software machine. The instruction set architecture is based on a **stack** with **activation records**. It was implemented in the 70s as the target architecture for Pascal compilers. Execution was by interpretation.

Modules

- PM/0 Virtual Machine
- Lexical Analyzer for PL/0
- Parser and Code Generator for tiny PL/0
- PL/0 Compiler

The lectures will present the

- theory
- data structures
- algorithms

necessary to implement an efficient compiler PL/0 => PM/0.