Intro. Computing with the C Programming Language

The Way of the Program

Shin Hong

31 August 2023

Programming

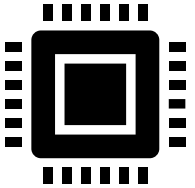
- Goal: Compose a sequence of computer instructions such that their executions offer a service that transform input to output
- Approach: gradually translate abstract description (high-level language) into concrete description (low-level language)

Natural language

- abstract: flexible, concise, portable
- informal: ambiguous and context-dependent

Formal language

- formal: concrete and actionable
- complex: detailed and machine-dependent



Binary: The Language of Computer Machine

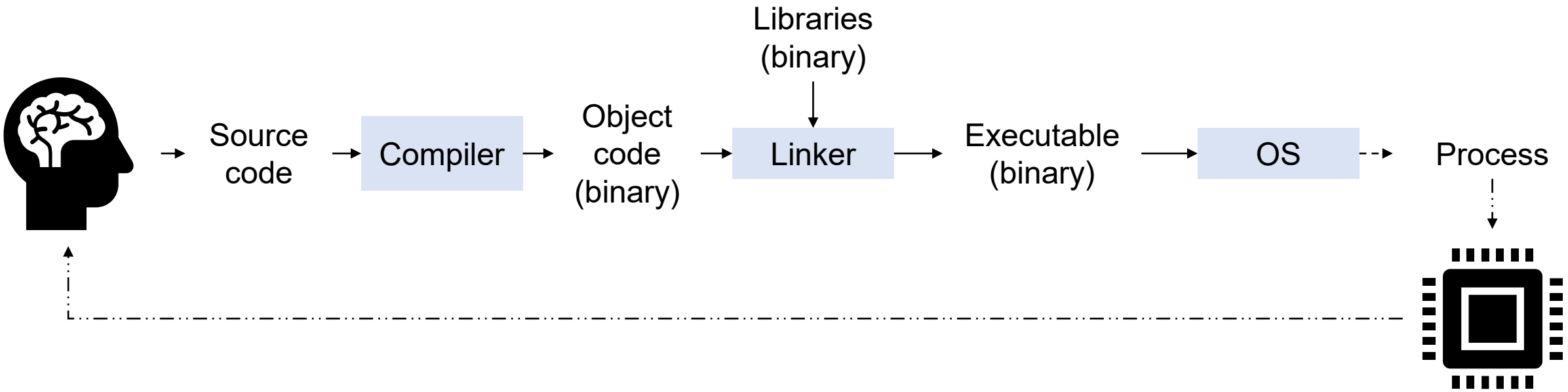
- written in a fixed set of instructions (commands), and a finite memory locations
  - such a system is called as a computer architecture
- binary code: the code that a computer machine can interpret directly
  - object code or executable
  - every piece must be given concretely
- challenges in programming binary
  - there are many families of computer architectures
  - the specific configuration of a computer may be different

Program Build Procedure (1/2)

- Take multiple translation steps to convert abstract command to concrete command by adding details and machine-dependency gradually

- Supporting system
  - Operating system (OS): offer high-level operations while hiding HW details
  - Programming language: provide a formal language for a human to write source code without considering HW details
  - Compiler: translate source code to binary code of a target machine
  - Linker: combine multiple binary code files into one executable program
  - Virtual machine (VM): offer a uniform computer architecture for all machines

Program Build Procedure (2/2)



Program

- A program is a finite sequence of instructions that specifies how to perform a computation
  - common kinds of instructions
    - input: receive signal from a device and produce internal data
    - output: transmit internal data to a display device
    - math: perform basic mathematical operation
    - testing: check for certain conditions for executing instructions
    - repetition: perform some instructions repeatedly

- Programming is the process of breaking a large, complex task up into smaller subtasks until the subtasks are simple enough to be performed with one of the instructions

Programming Errors

- Syntax refers to the rule about the structure of a program, and its violation is called as syntax error
  - compilers detect syntax errors, and verify the validity of program structure
- Computer systems rejects execution of an instruction when the instruction is not possible to be executed at the state
  - a runtime error occurs when an invalid instruction is given to be executed
  - e.g., divide-by-zero
- A program produces a wrong output if it has a logical error (semantic error)

First Program

```
#include <stdio.h>
#include <stdlib.h>

/* main: generate some simple output */

int main(void)
{
      printf("Hello, World.\n");
      return(EXIT_SUCCESS);
}
```

Intro. Computing with the C Programming Language

Variables and Types

Shin Hong

31 August 2023

Variants

```
int main(void)
{
printf("Goodbye, ");
printf("cruel world!\n");
return(EXIT_SUCCESS);
}
```

```
int main (void)
{
      printf ("Hello World.\n");      /* output one line */
int main(void){printf("Goodbye, ");printf("cruel world!\n");
return(EXIT_SUCCESS);}
}
```

current_time.c

```
#include <stdio.h>
#include <stdlib.h>

int main ()
{
        int hour = 3 ;
        int min = 15 ;
        char separator = ':' ;

        printf("The current time is ") ;
        printf("%d", hour) ;
        printf("%c", separator) ;
        printf("%d\n", min) ;

        return EXIT_SUCCESS ;
}
```

Variables and Values

- A variable is a named memory location that stores a value

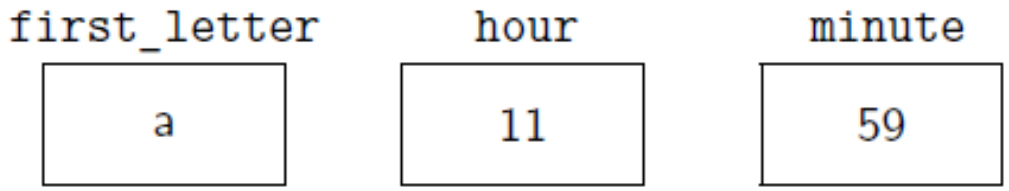
- In C, a variable is declared with a type which defines the kind of values that the variable can hold

```
int hour ;
int minute ;
char separator ;
double second ;
```

Assignment

- a statement to store a value to a variable
  - ex.

```
first_letter = 'a';    /* give first_letter the value 'a' */
hour = 11;             /* assign the value 11 to hour */
minute = 59;           /* set minute to 59 */
```



Variable Name

- most of words can be used as a variable name, except reserved keywords

Reserved keywords in the C language

auto	double	inline	sizeof	volatile
break	else	int	static	while
case	enum	long	struct	_Bool
char	extern	register	switch	_Complex
const	float	restrict	typedef	_Imaginary
continue	for	return	union	
default	goto	short	unsigned	
do	if	signed	void	

Operators

- Operators involves mathematical computations like addition and multiplication

```
1+1        hour-1        hour*60+minute      minute/60
```

- An expression is a combination of variables, values and operators, that defines a value

```
int hour, minute;
hour = 11;
minute = 59;
printf ("Number of minutes since midnight: %i\n", hour*60 + minute);
printf ("Fraction of the hour that has passed: %i\n", minute/60);

printf ("Percentage of the hour that has passed: ");
printf ("%i\n", minute*100/60);
```