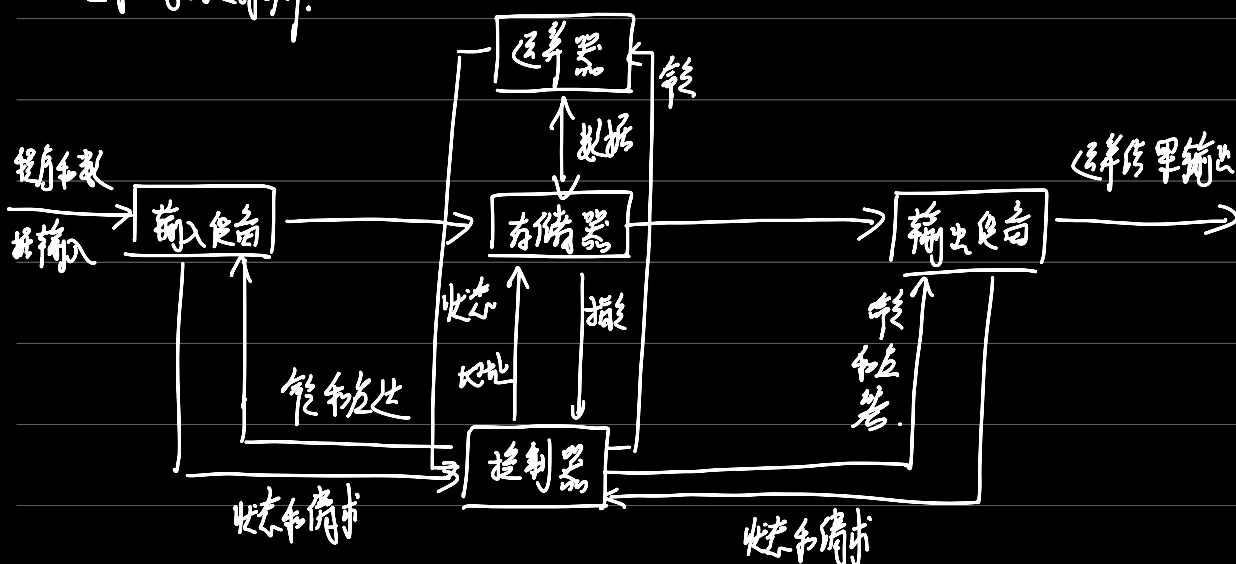
冯诺依曼结构　　存储程序

ENIAC：第一台通用计算机（十进制表示信息）

主存

部件 → 取出指令

是来执行指令的部件.



存储器不仅能存放数据也能存放指令

控制器自动取出指令执行

运算器进行加/减/乘/除操作，和逻辑运算，附加运算.

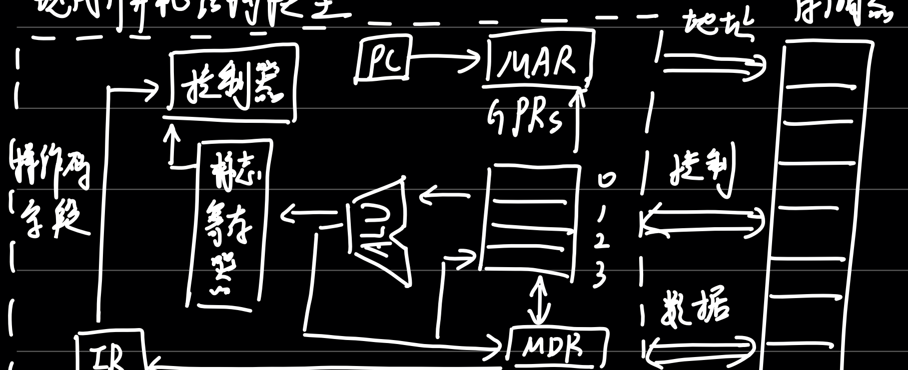
操作人员通过输入设备，输出设备和主机通信.

内部以二进制表示指令和数据

每条指令由操作码和地址码组成

主存的地址寄存器

现代计算机结构模型



划分成一个一个存储单元

CPU: central process unit 中央处理器

PC: program counter 程序计数器

MAR: memory address register 存储器地址寄存器

ALU: Arithmetic Logic unit 算术逻辑部件.

IR: Instruction register 指令寄存器

MDR: Memory data register 存储器数据寄存器.

GPRs: General purpse register set 通用寄存器组.

计算机如何工作?

程序执行前

数据指令输入存储器, 每条指令和每个数据都有地址, 指令按序放, 由OP. ADDR字段组成, 程序起始地址放入PC.

开始执行程序.

①根据PC取指令 ②指令译码 ③取操作数 ④指令执行 ⑤回写结果 ⑥修改PC的值 继续执行下一条指令

指令格式提供信息:

① 操作性质(操作码)

② 源操作数1或1和源操作数2 (立即数, 寄存器编号, 存储地址)

③ 目的操作数地址 (寄存器编号, 存储地址)

程序开发过程

① 机器语言编写程序，穿孔为0，未穿孔为1 — 输入：按钮、开关

书写、阅读困难 — 输出：指示灯等。

② 汇编语言

用符号表示跳转位置和变量位置。

- 助记符表示操作码
- 标号表示位置
- 助记符表示寄存器

不用手动计算指令所需要的地址代码

不需人工指令编码，编写方便。

可读性比机器语言强。

汇编语言 —(汇编程序)→ 机器语言

面向动作

在不同结构的机器上无法运行。

机器/汇编语言都是面向机器结构的语言，称为机器级语言。

③ 高级语言

面向算法描述

编译：将高级语言源程序转化为机器级目标程序，执行时只需启动目标程序即可。

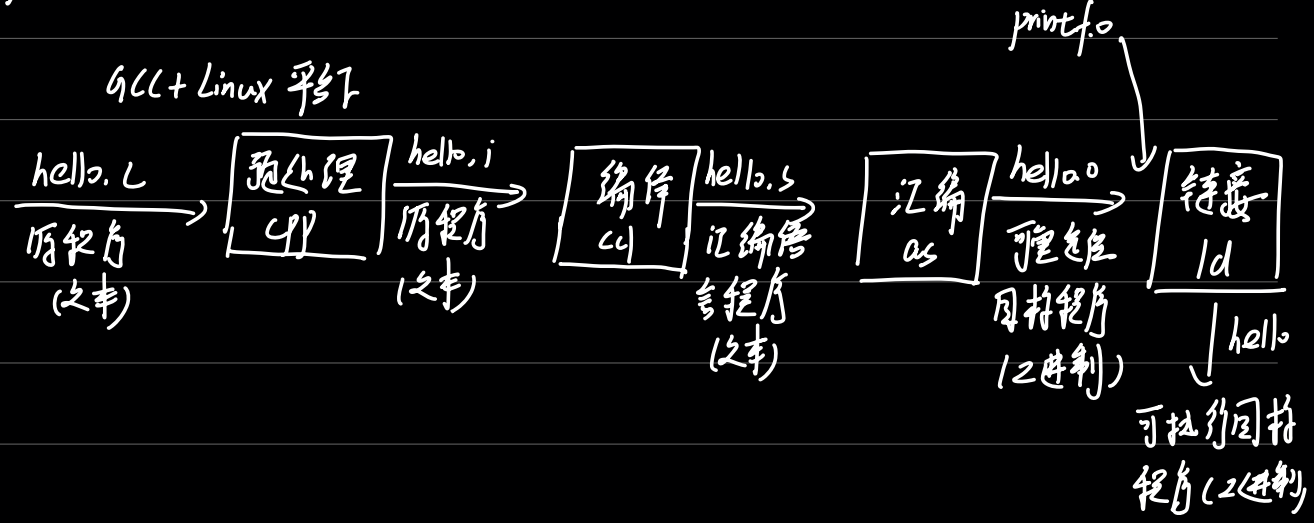
解释：将高级语言程序逐条翻译成机器指令并执行，不生成目标文件。

一个典型程序的转换处理过程

源程序中存放字符的ASCII码。

```
#include <stdio.h>
int main () {
    printf("hello, world\n");
    return 0;
```

]



支撑程序开发和运行的底层系统软件提供，包括操作系统和操作系统

早期计算机系统层次

最早的计算机用机器语言编程.

第一代程序员什么语言.

应用程序

操作系统

计算机硬件.

汇编语言

应用程序

汇编程序

操作系统

计算机硬件

和机器交互的界面

高级语言

第三代程序员什么语言为过程式语言.

应用程序

语言处理系统

描述实现过程 / 操作系统

第四代程序设计语言为非过程化语言，编程时只需说明做什么 / 指令集体系结构

计算机硬件.

语言处理系统：各种语言处理程序（编译、汇编、链接）、运行时系统（如库函数、调试、优化等功能）

操作系统包括人机交互界面、提供服务功能的内核例程

计算机系统抽象层转换

- 应用 —— 下层是上层实现，上层是下层抽象
- 程序员 {
  - 算法
  - 编程
  - 操作系统/虚拟机
- 架构师 {
  - 指令集体系结构 ⟶ 对底层硬件的抽象，软硬件交界的接口.
  - 微体系结构
  - 功能部件
- 电路
- 器件.

计算机系统的不同用户

最终用户：工作在由应用程序提供的最上面的抽象层

系统管理员：工作在由操作系统提供的抽象层

应用程序：工作在由操作系统（主要有编译器和汇编器）的抽象层

系统软件：工作在ISA层之上，包括系统软件。

ISA 非重要 instruction set architecture

规定软件如何使用硬件。

- 可执行指令集合，包括指令格式，操作种类以及每种操作对应的操作数
- 指令可接受的操作数类型
- 操作数所能存放的寄存器组的结构，包括寄存器名称、编号、长度和用途
- 操作数所能存放的存储空间大小和编址方式。
- 大端/小端
- 寻址方式
- 指令执行过程的控制机制，包括PC、手作码定义等。