**프로그래밍 언어 과제1**

**20161595**

**배성현**

**1. 일반적으로 FORTRAN 언어는 과학기술용 언어라고 하고, COBOL 언어는 비지니스용 언어라고 한다. 여기서 "~용 언어"의 의미를 설명하고, 최근 Python언어가 딥러닝과 같은 인공지능 프로그래밍용 언어로 불리우는 이유를 설명하라.**

“~용 언어”라는 의미는 특정 목적을 위해서 필요한 기능들을 해당 언어가 많이 가지고 있다는 의미이다. 즉 특정 기술을 표현하는 데에 있어서 꼭 필요한 기능들을 언어에서 쉽게 표현해준다. 예를 들어 FORTRAN 언어는 과학기술을 표현하는 데에 필요한 기능들을 많이 가지고 있고 COBOL 언어는 input, output을 정밀하게 제어하기 위한 기술들을 많이 가지고 있기 때문에 비즈니스용 언어라고 한다. 최근 Python언어가 딥러닝과 같이 인공지능 프로그래밍용 언어로 불리는 이유는 Python언어가 딥러닝을 하는데 필요한 머신러닝 라이브러리와 프레임워크를 지원하기 때문이다. 여러 머신러닝 및 딥러닝 프레임 워크들의 대부분이 Python의 영역에 있거나 Python API를 지원한다. 또한 Python 언어는 딥러닝 모델을 테스트를 하거나 온라인으로 서비스를 제공하는 등의 작업들을 하는데 필요한 풍부한 도구를 제공한다. 따라서 Python 언어를 인공지능 프로그래밍 언어라고 부를 수 있다.

**2. 프로그래밍 언어는 기계어, 어셈블리어, 고급언어 등과 같은 계층적인 구조를 가지고 있다. 이런 계층적인 구조를 가지게 된 이유를 설명하고, 각각 언어의 용도를 설명하라.**

프로그래밍 언어는 프로그래머가 하드웨어를 어떻게 동작시킬 지에 대해서 명령을 내릴 때 사용하는 언어이다. 처음에는 컴퓨터의 CPU가 직접 해독하고 실행할 수 있는 비트단위로 쓰인 기계어를 사용하였다. 하지만 기계어의 0,1로 연산을 표현하고 이해하기에는 너무 힘들었기 때문에 이를 조금 더 심볼릭하게 표현하는 어셈블리어가 생겼다. 또 어셈블리어는 기계어와 1대1로 매칭되어 표현되었기 때문에 연산을 표현하고 이해하기에 코드가 너무 길었다. 따라서 코드를 줄이고 더 심볼릭하게 표현할 수 있는 고급언어가 생기게 되었다. 이렇게 프로그래밍 언어가 계층적인 구조를 가지게 되었다.

기계어나 어셈블리어로 프로그래밍을 하게 되면 프로그램의 속도가 빠르고, 프로그램의 크기도 작아지게 된다. 따라서 기계어나 어셈블리어는 파일 크기가 작으면서 실행속도가 빨라야 하는 프로그램이나 고급언어로 작성하였을 때 속도가 느려서 프로그램이 제 역할을 하지 못하는 경우에 사용한다. 고급언어의 경우 공통적인 특징으로는 사람의 입장에서 읽기, 쓰기, 유지보수가 쉽다는 것이다. 따라서 고급언어는 사람이 이해하기 쉽게 사람의 입장에서 프로그래밍을 할 때 사용한 다고 할 수 있다. 또한 세부적으로 들어가게 되면 고급언어는 각 언어들마다 갖고 있는 특징과 제공하는 기능들이 다르기 때문에 각 언어들마다 용도가 다르다고 할 수 있다.(FORTRAN:과학기술용, COBOL:비즈니스용, PASCAL:교육용 등)

**3. 32-bit CPU 혹은 16-bit CPU를 구별하는 가장 큰 요소는 무엇인가? 또한, 32-bit CPU는 어떤 부분이 32-bit인지 모두 나열 하고 설명하라.**

32-bit CPU와 16-bit CPU를 구별하는 가장 큰 요소는 한 머신 사이클에 처리할 수 있는 비트의 수이다. 즉 32-bit CPU는 32비트 인풋 두 개를 받아서 한 머신 사이클에 32비트 계산을 할 수 있으며 16-bit CPU는 16비트 인풋 두 개를 받아서 한 머신 사이클에 16비트 계산을 할 수 있다. 또한 몇 비트 CPU이냐에 따라서 기본 단위인 word사이즈가 달라지게 되고 16-bit CPU의 word사이즈는 2바이트, 32-bit CPU의 word사이즈는 4바이트이다. 32-bit CPU는 레지스터의 크기, 주소 버스의 크기, 데이터 버스의 크기가 32-bit이다. 레지스터는 컴퓨터의 프로세서 내에서 자료를 보관하는 아주 빠른 기억 장소이고 계산을 수행 중인 값을 저장하는 것에 사용되는데 데이터, 주소, 범용, 부동 소수점, 상수, 특수 레지스터 등이 있으며 32-bit CPU에서는 이러한 레지스터들의 크기가 32비트 이다. 주소 버스는 데이터의 근원지 또는 목적지의 일정한 메모리 주소를 전달하는 데에 사용되고 32-bit CPU는 32개의 주소 버스를 가지게 된다. 데이터 버스는 시스템 모듈들 간의 데이터 이동 경로를 제공하게 되는데 32-bit CPU에서는 32개의 비트를 데이터 버스를 통하여 전송할 수 있다.

**4. CISC와 RISC가 등장하게 된 배경을 설명하고, 각 종류에 대한 대표적인 CPU를 조사하여라**

프로그래밍 언어는 기계어에서 어셈블리 언어, 고급언어로 발전하게 되었다. 그리고 고급언어로 프로그래밍을 하다 보니 Class나 Array 같은 data structure나 여러 기능들이 필요하게 되었고 이러한 data structure나 기능들을 핸들링 하는 머신을 하드웨어에 추가하게 되었다. 이러면서 CPU가 복잡해지게 되었고 이 것이 CISC이다. CISC는 일련의 명령어를 순차적으로 처리하기에 좋고, CPU의 동작 속도와 성능이 비례한다. 대표적인 CISC 프로세서로는 x86(인텔 8086, 80486 등), 펜티엄이 있다. 하지만 CISC는 명령어의 개수나, 형식으로 보았을 때 명령어가 복잡하고 명령어 길이가 가변적이었다. 따라서 자주 사용되는 명령어는 짧은 명령어 형식이었고 상대적으로 긴 명령어 형식은 덜 사용되었다. 또한 한 명령어 내의 피 연산자들이 다양한 주소방식을 사용하는 것을 허용하여 명령어의 인출과 해독이 복잡하였다. 또한 회로가 복잡하여 칩 크기가 커지게 되었다. 이러한 CISC프로세서의 단점들 때문에 나오게 된 것이 RISC프로세서이다. RISC는 명령어 길이를 고정하여 인출과 해독을 단순화 하였고, 제한된 개수의 명령어를 사용하여 논리 회로를 단순화하였다. 회로를 단순화 함으로써 칩의 면적이 감소하였고 여유공간에 많은 수의 범용 레지스터를 사용하여 메모리 접근을 최소화했다. 대표적인 RISC방식의 CPU들은 ARM, MIPS, POWER, RISC-V가 있다.