

2024 10 22 수업

☐ 10.23(내일) 휴무, 11.20 11.21 11.22 휴무

☐ project - 11월 중순에 시작될 예정, B반이랑 같이 프로젝트 같이 할 작정(같이 안할 가능성이 높음)

☐ 🍒 리눅스 접속 시 "호스트에 연결 할 수 없습니다" 라고 뜸:해결)) 컴퓨터 지문 인식
.ssh>known-host(지문) 지우기🍒
**config는 내 host 이름이랑 ip 주소 등

☐ 🎉🍰 numpy 수업 🎉🍰



print(0.1+0.2)하면 0.3이 나올까?

ChatGPT의 말:

ChatGPT

`print(0.1 + 0.2)` 를 실행하면 0.3이 아닌 **0.30000000000000004**가 출력됩니다.
이는 **부동소수점 연산의 특성** 때문에 발생합니다.

컴퓨터는 소수를 이진수로 변환하여 저장하는데, 0.1과 0.2 같은 소수는 이진수로 정확하게 표현되지 않아 약간의 오차가 생깁니다. 이런 오차가 합산 결과에 반영되어, 기대한 0.3 대신 0.30000000000000004와 같은 결과가 나타납니다.

이 문제는 대부분의 프로그래밍 언어에서 공통적으로 발생하며, 필요한 경우에는 소수점 계산에서 **반올림**이나 **정밀도 제어**를 통해 해결할 수 있습니다.

■ 32bit, 64bit → bit란?

bit: 2진법일때 자리 갯수 ex) 2bit _ _ ⇒ 2자리수 의미

byte: 컴퓨터가 처리하는 최소단위, bit가 8개

Q) 컴퓨터는 0과 1을 읽는데 왜 byte가 최소단위인가?

A) 1bit 0-1(2개), 1byte 2의 8제곱 개 표현 → 명령어

bit로 읽으면 너무 작아서 명령어를 만들 수 있는 최소크기인 byte를 사용한다.

■ ex) 1byte = 1개 문자(영어) = 8bit(문자(영어)를 전부 표현 가능 a, ab, abc 모두 표현 가능(8bit안에서))

그러나 한글은 3byte가 필요

■ ex) 1byte _ _ _ _ _ ⇒ 8자리 = 8bit 2의 8제곱 개의 표현 = 명령어 개수가 2의 8제곱 개라는 뜻

■ ex) 32bit = 4byte

32 bit 운영체제는 32자리 즉 2의 32제곱 개의 명령어가 있다.

32 bit 운영체제(4GB) ⇒ 는 일종의 방(필요공간)이다. 방안에 실제로 들어있는 것은 = 명령어 개수(제조사 마다 다름)

■ 용량의 단위

bit(2의 1제곱)-byte(2의 4제곱)-KB(2의 10제곱)-MB(2의 20제곱)-GB(2의 30)-TB(2의 40)-PB(2의 50)-EXA(2의 60)

Q) 32 bit 운영체제의 명령어 총 용량은?

A) 2의 32제곱 개 이다. why) 2의 2제곱 X 2의 30제곱 = 4 X 2의 30제곱 = 4GB (2의 30=GB)

■ 메모리량 명령어 개수의 관계

명령어 개수 만큼 CPU가 최대로 처리 가능, 최대 쓸 수 있는 메모리는 명시적

Q) 64bit 운영체제의 명령어 총합은?

A) 2의 64제곱 개 이다. why) 2의 64 = 16exa바이트 만큼 메모리 인식

cf) 같은 32bit 인데 risc(2의 10제곱), cisc(2의 20제곱, 보통 cisc 많이 씀) 나뉜다 왜 그럴까? A) 나누는 이유는 명령어가 적으면 cpu회로가 적다 ⇒ 즉 전력도 적다 오래가지만 성능이 낮다(다른 명령어 처리 못하고 성능이 느리다.)

/

■ 2진수 표현방법

(부호) significand(부동) X 2의 exponent제곱

■ 부동 소수점 연산

7514 \Rightarrow 7.514 X 10의 3제곱 (7.514:coefficient(계수), 10:basenumber(진수), 3:exponent(지수))

여기서 바뀌는 값은? coefficient(계수), exponent(지수)가 바뀜

$1 \leq \text{coefficient} < \text{basenumber}$

■ 부동(float) \Rightarrow why? 움직이지 않는다. 떠다닌다(부력) \Rightarrow floating point(부동 소수점)

왜 떠다닌다고 표현할까? 점이 움직이니깐 ex) 123.45 \Rightarrow 12.345

■ 부호(sign) X 가수(significand) X 밑수(basenumber) (지수(exponent))제곱

\Rightarrow 실수의 표현방법

ex) 1234 \Rightarrow 1 X 1.234 X 10의 3승 \Rightarrow why) 2진법 표현 용이해서

$1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 \Rightarrow$ 정규화 들의 덧셈

★ 정규화

$0.375 \times 2 \Rightarrow 0.001(2) \Rightarrow 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} + 0 \times 2^0$

\Rightarrow 컴퓨터는 어떻게 표현할까? (IEEE규격)

1bit 부호

8bit 지수

23bit 가수

부호 X 가수 X 밑수의 (지수)제곱

