

Xilinx Zynq FPGA, TI DSP, MCU 기반의 프로그래밍 및 회로 설계 전문가 과정

12일차 숙제 – (자료구조)

강사 – **Innova Lee(이상훈)**
gcccompil3r@gmail.com

학생 – **안상재**
sangjae2015@naver.com

* 사전평가 문제

2. Q : Tree 는 Stack 이나 Queue 와는 다르게 어떠한 이점이 있는가 ?

A. Stack 과 Queue는 노드의 구조가 일렬로 되어서 데이터를 삽입할 때에 처음 노드부터 맨 끝의 노드까지 이동해야하고, 삭제를 할 때에도 처음 노드부터 그 다음 노드로 하나씩 이동해서 삭제할 노드를 검색해 나가야 한다. 즉, 노드의 구조가 직렬화 되어 있어서 삽입과 삭제 시 상당히 비효율적이다.

반면, Tree는 노드 구조가 직렬화 되어 있지 않고, 나무의 형태를 띄워서 Stack 과 Queue보다는 검색이 빠르다는 장점이 있다.

4. Q : int p[7] 과 int (*p)[7]의 차이는?

1) int p[7] : int형의 데이터 7개를 저장할 수 있는 배열 p

2) int (*p)[7] -> int (*)[7] p 로 바꿀 수 있다. : int 형을 7개 저장하는 배열의 포인터

6. Q : Intel Architecture 와 ARM Architecture의 차이는?

1) ARM은 Load/Store Architecture 구조이지만 x86은 그렇지 않다.

2) x86 Architecture은 stack이 -방향으로 자라지만, ARM은 stack이 +방향으로 자란다.

7. Q : 이것이 없으면 C 언어의 함수를 호출할 수 없다. 이것은?

A : C 프로그램은 main함수가 동작하는 것이다. C 언어의 모든 함수는 main함수 안에서 호출된다. main함수 하나가 하나의 프로세스라고 보면 된다.

9. Q : Memory Hierarchy(메모리 계층 구조)에 대해 기술하시오.

A : 메모리는 속도가 빠른 순서로 레지스터-캐시-메모리-디스크 이고, 가격이 비싼 순서는 역순이다.

10. Q : Memory Hierarchy(메모리 계층 구조)에 대해 기술하시오.

A : 메모리는 Stack, Heap, Data, Text의 4영역으로 나뉘어진다.

1) Stack : 지역 변수가 저장되는 영역

2) Heap : 동적 할당된 변수가 위치하는 영역

3) Data : 전역 변수 및 static으로 선언된 변수들이 위치하는 영역. 이 영역에서 초기화 되지 않은 것들은 모두 0으로 저장됨.

4) Text : 기계어 code가 위치하는 영역

11. Q : 파이프라인이 깨지는 경우에 대해 기술하시오.

A : if문, switch문, goto문, continue문과 같이 jmp 명령어를 수행하는 경우에 파이프라인이 깨지게 된다.

12. Q : void (* signal(int signum, void (* handler)(int)))(int)라는 signal 함수의 프로토타입을 기술하시오. 프로토타입을 기술하라는 의미는 반환형(리턴 타입)과 함수의 이름, 그리고 인자(파라미터)가 무엇인지 기술하라는 뜻임.

A :

-> void (*)(int) signal(int signum, void (* handler)(int)) 으로 변형할 수 있다.

1) 반환형 : int형을 인자로 갖고 void형을 반환하는 함수의 포인터

2) 함수이름 : signal

3) 인자 : int 형 데이터 signal, int형을 인자로 갖고 void형을 반환하는 함수의 포인터 handler

13. Q : goto 를 사용하는 이유에 대해 기술하시오.

A : goto 아래의 코드를 실행하지 않고, 원하는 코드로 이동해야 하는 경우 사용함.

22. Q : AVL 트리는 검색 속도가 빠르기로 유명하다.

Red-Black 트리도 검색 속도가 빠르지만 AVL 트리보다 느리다.

그런데 어째서 SNS 솔루션등에서는 AVL 트리가 아닌 Red-Black 트리를 사용할까 ?

A : AVL 트리는 검색속도는 빠르지만, 데이터의 삽입이나 삭제를 할 때마다 트리의 균형을 맞추기 위해 노드 구조를 재조정해야 한다. 노드가 수십만, 수백만, 수천만개가 될 경우 노드 구조를 재조정하는 작업이 시간을 굉장히 많이 소요하기 때문에 데이터의 삽입과 삭제가 매우 빈번하게 수행되는 분야에서는 AVL 트리가 적합하지 않다.