System programming 3분반

기말고사 32193430 이재원

1.

1.1 label, opcode, operand

1.2 AF

1.3 global directive

1.4 SIMD

1.5 make

1.6 blktrace

1.7 lscpu, readelf

1.8 ① cpu에서 메모리에 접근하기 효율적이다.
② cache line에 잘 맞춰서 cache 효과가 향상된다.

1.9 pipeline은 여러 명령어를 동시에 수행하는 것이다.

1.10 전통적으로는 CPU cache가 더 작고 빠른데, ~~그러면 자신보다 높은 계급을~~ .chache Effect 같은 것이 달라지지 않나? 굳이 그렇게 해야 하는 의문이 있지만, 미래는 모르니까 이미 누군가는 만들었을 수도 있다.

2.

2.1 Func(2024, 2023, 2022, 2021)
a=2024, b=111, c=3, d=116, e=40

ebx=2 ~~024~~ 518
eax=111
ecx=~~8~~3

stack frame
2021
2022
2023
2024
return add.
saved ebp ← ebp

2.2 .text : text section 선언
.long : 4bytes
.push : stack에 데이터 or register 넣는다.
.leave : 함수 종료하고 stack Frame 내에 있는 거 종료.

2.3 int : 시스템 호출, 커널 모드, eax에 시스템 호출 번호 넣어야 한다.
call : 함수 호출, 사용자 모드, call할 때 함수 주소 지정

2.4 8과 %esp 내부의 데이터가 1, 1 이면 1, 0, 0 이면 0,
1, 0 이면 0 과 같은 식으로 and 연산을 수행한다.

3.1 movl %esp, %ebp : register operand
movl 8(%ebp), %ebx : memory operand : Base plus offset addressing

3.2 ① movl 8(%ebp), %ebx
② movl %ebx, a) data dependency → data hazard.

① Ifet Dec Dfet Exe Res
② Ifet Dec Dfet Exe Res
2번의 stall 발생!

3.3 instruction reordering, out-of-order execution.

3.4

3.5 NUMA : Non-Uniform Memory Access 의 약자.
아래 프로그램에서 메모리에 접근할 때 시간이 균일하지 않을 수 있다.

4.1 code motion : 반복문 내 불필요한 것을 외부로 이동하여 성능 향상.

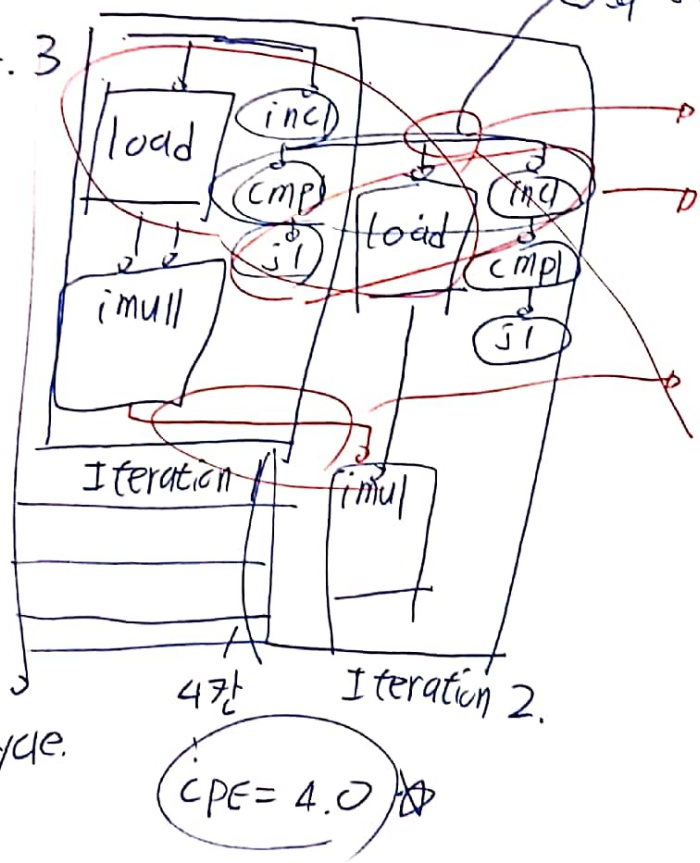
Loop unrolling : 한 반복문에서 연산 여러번 수행하여 CPE 낮추어 성능 향상.

Loop splitting : loop 내에서 변수 2개 이상 써서 주로 곱셈 성능 향상

4.2 combine3()는 register indirect addressing으로 메모리에 접근하는 데 시간이 걸리지만,

combine4()는 register addressing으로 register만 접근하여 성능이 따른

4.3



① superscalar : 동일한 정수형 연산 동시 수행

② pipeline : load 끝나기 전 다음 load 수행

③ branch prediction.

④ data dependency

⑤ out-of-order execution.

4.4 c=a+b

O : movl a, %eax
addl b, %eax
movl %eax, c

O3 : movl $4, 4
↳ 하나의 명령어로 처리하기 때문에 빠르다.

5.1 ① Scanner ② parsing : 문법 분석 ③ Semantic Analysis 의미분석
④ Code generation : 코드 생성.

5.2

ba 13 12 21 20
89 e6
8b 38

답.

8 4 2 1 4 2 1
1 1 1 0 0 1 0 1
14 6

21
0 0 1 1 1 0 0 0
3 8

10 A
11 B
12 C
13 D
14 E

6.1 $S = \frac{1}{(1-a) + d/k}$ ·성능향상.

a 비중

많이 차지하는 부분을 성능 향상 시켜야 한다!