

Blog2Book, 프로그래머가 몰랐던 멀티코어 CPU 이야기
1쇄 오탈자 목록, 2010년 8월 8일

범례 오탈자 중요도에 따라 별표(★)를 매겼습니다. 빈 칸은 사소한 오탈자 및 어색한 문장을 뜻합니다.

| 분류 | 장 | 페이지 | 위치 | 내용 |
|-----|----|-----|------------------------|--|
| | 1 | 18 | 아래에서 6번째 줄 | 40년이 지나 → 30년이 |
| | 2 | 23 | 5번째 줄 | 프로그래머가 실수로 문법에 → 문법이 |
| ★★ | 2 | 37 | (5) 설명 | r2가 이 값을 가진다 → r3이 이 값을 가진다 |
| | 3 | 55 | 맨 아랫줄 | 마치 ... 작동한다고 배웠다 → 마치 ... 작동하는 것처럼 보인다고 배웠다. |
| | 3 | 59 | 그림 3-5 | 그림 번호 수정: 그림 3-5 → 그림 3-3 |
| ★★ | 4 | 64 | 수식 4-1 | S: 성능 향상(스피드업) → S: 최적화된 부분의 성능 향상 |
| | 4 | 67 | 첫번째 줄 | 프로세서/스레드 사이에 → 프로세스/스레드 사이에 |
| ★★ | 4 | 69 | 수식 | $T = (0.5*1 + 0.2*2 + 0.1*3 + 0.2*4)(\text{사이클/개})$ * 1,000,000,000(개) * 1/2,000,000,000(초/사이클) = 1초 |
| | 5 | 75 | 첫번째 줄 | 프로그래머는 자기가 만든 프로그램을 당연히 프로세서간 |
| ★★ | 5 | 81 | 2번째 단락 3번째 줄 | 컴파일러가 어떤 주소 값을 가리킬지 소스 분석으로 ... [부연설명] 컴파일러가 두 메모리 연산 사이의 의존성을 예측하는 것은 일반적으로 매우 어렵습니다. 그 이유는 의존성 여부를 판단하려면 결국 두 메모리 연산이 가리키는 주소 값의 범위를 알아야 하는데, 코드를 실행하지 않고서는 이것을 알기가 굉장히 어렵기 때문입니다. |
| | 6 | 88 | 소스 6-1 | 2: fetech() → fetch() |
| | 6 | 99 | 2번째 단락 7번째 줄 | 보통 예외(exception)과 → 보통 예외(exception)와 |
| | 6 | 101 | 소스 5-8 | 소스 5-8 → 소스 6-8 |
| | 7 | 106 | 첫번째 단락 마지막 줄 | 처리율리 → 처리율이 |
| ★ | 7 | 123 | 2번째 단락 첫번째 문장 | Ferret은 주어진 ... → Ferret은 주어진 디렉터리나 데이터베이스에 있는 이미지를 이미지 내용을 이용하여 서로 비교하는 프로그램이다. |
| | 7 | 125 | LoadThread 함수 내 | queue_enqueue_wait(&q_load_seq, data); → load |
| ★★★ | 8 | 141 | 그림 8-5 | 명령 2 : r2 = r0 + 1; → 명령 2 : r1 = r0 + 1; |
| ★★★ | 8 | 142 | 소스 8-3 Line 3, 4 주석 | r0를 F0 → r0를 F1 |
| ★★★ | 8 | 143 | 첫번째 단락 2번째 줄 | 여기서는 F2를 얻었다 → 여기서는 F0를 얻었다 |
| ★★★ | 8 | 143 | 첫번째 단락 5번째 줄 | ~ 물리 레지스터 F0으로 바꾸었다. → ~ 물리 레지스터 F1로 바꾸었다. |
| ★★ | 9 | 153 | 대화문 밑 단락 | [부연설명] 하이퍼스레딩과 동시 멀티스레딩 기술이 약간 혼동되어 쓰였습니다. 하이퍼스레딩은 동시 멀티스레딩 기술의 한 구현의 예입니다. 인텔 펜티엄4/Core i7의 하이퍼스레딩은 두 개의 논리 프로세서를 만들지만 두 개 이상도 가능하다는 설명을 쓰려다가 단어 선택에 약간 혼란이 있었습니다. |
| | 9 | 158 | 9번째 줄 | Swith-On-Event → Switch-On-Event |
| | 10 | 167 | 2번째 단락 7번째 줄 | 불과 몇 픽셀의 단색 모니터에 → 몇 백 픽셀의 |

Blog2Book, 프로그래머가 몰랐던 멀티코어 CPU 이야기
1쇄 오탈자 목록, 2010년 8월 8일

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----------|----|----------|--|--|--------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|----|----|----|---|---|---|----|-----------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|----|--------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|----|------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ★ | 10 | 174 | 3번째 단락 3번째 줄 | 윈도라는 명령어 개념을 → 명령어 윈도우라는 개념을 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 10 | 177 | 2.의 3번째 줄 | [조사탈락] 구조에 따라 데이터를 주고받는 논리적인 형태가 다르다. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ★★ | 10 | 180 | 첫번째 단락 4번째 줄 | 많은 이들에게 익숙한 멀티스레드 방법론이 적용된다. [부연설명] 이미 많은 개발자가 하고 있는 일반적인 멀티스레드 프로그래밍 방법론이 여기에 해당된다. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 10 | 186 | 4번째 줄 | nVidia의 ... 그래픽 카드는 160GB/s의 대역폭이 엄청나다. → NVIDIA의 ... 그래픽 카드는 대역폭이 160GB/s이 될 정도로 엄청나다. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ★ | 10 | 188 | 둘째단락 첫째줄 | 그림 10-7 → 그림 10-6 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | 189 | 첫번째 줄 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 11 | 191 | | NVIDIA사의 사명을 nVidia로 적었습니다. 공식적으로 NVIDIA라고 씁니다. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 11 | 206 | 소스 11-4 | 변수명 N → size | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 11 | 207 | 소스 11-5 Line 10 | for (int k = 0; k < <u>Width</u> ; ++k) → N | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 11 | 209 | 소스 11-6 Line 9, 11 | 변수 gpuA, gpuB, gpuC는 float* 형인데 float**으로 가정하고 적었습니다. 예를 들어, gpuA[row][k] → gpuA[row*N + k] | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 12 | 218 | 3번째 줄 | 이 루프는 또한 <u>data[0]</u> 부터 <u>data[N-1]</u> 까지 순차 접근하므로 공간 지역성도 있다. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 12 | 224 | 둘째 문단 첫번째 줄 | [조사탈락] 캐시 라인 하나를 할당하는 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ★★★ | 12 | 229 | 두번째 줄 | 그 결과 <u>L1캐시</u> 로의 접근 패턴은 시간적 지역성이 → <u>L2캐시</u> 로의 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ★★★ | 12 | 230 | 박스 내 공식 | 미스 비율(miss rate = 캐시 히트 / 캐시 미스) → 미스 비율(miss rate = 캐시 미스 / (캐시 미스 + 캐시 히트)) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 13 | 242 | 아래에서 2번째 줄 | 분기 예측에 대해 이야기하기 전에 분기문의 → 분기문을 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 13 | 245 | 첫번째 문단 아래 2째줄 | 7-E은 → 7-E는 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ★★★ | 13 | 246 | 그림 13-2 | 100번지에 저장된 값, <u>1004</u> 로 분기한다 → ... <u>20</u> 로 분기한다 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ★★★ | 13 | 250 | 소스 13-3 | 1: if(i == 10) goto 3; → goto 4; | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ★★★ | 13 | 250 | 아래에서 3번째 줄 | 이 코드에서 분기문이 1이 조건을 만족해 <u>3번으로 간다면</u> → 4번으로 간다면 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ★★★ | 13 | 254 | 그림 13-7: 정확도 OX 표시에 오류가 있습니다. 매 루프 순환 7 마다 정확도는 X입니다. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | <table border="1"> <tr> <td>루프 순환:</td> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td> <td>0</td><td>..</td> </tr> <tr> <td>실제 분기 결과:</td> <td>T</td><td>T</td><td>T</td><td>T</td><td>T</td><td>T</td><td>T</td><td>N</td> <td>T</td><td>T</td><td>T</td><td>T</td><td>T</td><td>T</td><td>N</td> <td>T</td><td>T</td><td>T</td><td>T</td><td>T</td><td>T</td><td>T</td><td>N</td> <td>T</td><td>..</td> </tr> <tr> <td>분기 예측:</td> <td>N</td><td>T</td><td>T</td><td>T</td><td>T</td><td>T</td><td>T</td><td>T</td> <td>T</td><td>T</td><td>T</td><td>T</td><td>T</td><td>T</td><td>T</td><td>T</td><td>T</td> <td>T</td><td>T</td><td>T</td><td>T</td><td>T</td><td>T</td><td>T</td><td>T</td> <td>T</td><td>..</td> </tr> <tr> <td>정확도:</td> <td>x</td><td>o</td><td>o</td><td>o</td><td>o</td><td>o</td><td>o</td><td>x</td> <td>o</td><td>o</td><td>o</td><td>o</td><td>o</td><td>o</td><td>o</td><td>x</td> <td>o</td><td>o</td><td>o</td><td>o</td><td>o</td><td>o</td><td>o</td><td>x</td> <td>o</td><td>..</td> </tr> </table> | | 루프 순환: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | .. | 실제 분기 결과: | T | T | T | T | T | T | T | N | T | T | T | T | T | T | N | T | T | T | T | T | T | T | N | T | .. | 분기 예측: | N | T | T | T | T | T | T | T | T | T | T | T | T | T | T | T | T | T | T | T | T | T | T | T | T | T | .. | 정확도: | x | o | o | o | o | o | o | x | o | o | o | o | o | o | o | x | o | o | o | o | o | o | o |
| 루프 순환: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | .. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 실제 분기 결과: | T | T | T | T | T | T | T | N | T | T | T | T | T | T | N | T | T | T | T | T | T | T | N | T | .. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 분기 예측: | N | T | T | T | T | T | T | T | T | T | T | T | T | T | T | T | T | T | T | T | T | T | T | T | T | T | .. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 정확도: | x | o | o | o | o | o | o | x | o | o | o | o | o | o | o | x | o | o | o | o | o | o | o | x | o | .. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ★★ | 13 | 255, 262 | 소스 13-6, 13-8 | Line 11과 12에서 sizeof(predictor_entry_size) → sizeof(bool)*size_ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ★ | 14 | 270 | 첫번째 줄 | 그러나 소스 14-2에서 → 그러나 소스 14-1에서 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ★ | 14 | 270 | 아래에서 6번째 줄 | Compute의 ... 4번 또는 7번이 → Compute의 ... 2번 또는 4번이 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ★ | 14 | 272 | 3번째 줄 | <u>소스 14-3</u> 에 적어보았다 → <u>소스 14-2</u> 에 적어보았다 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ★ | 14 | 273 | 아래에서 7번째 줄 | <u>소스 14-3</u> 의 4,5,6은 → <u>소스 14-2</u> 의 4,5,6은 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 14 | 275 | 소스 14-4 | 1: dir->dir->type → dir->tdir_type | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Blog2Book, 프로그래머가 몰랐던 멀티코어 CPU 이야기
1쇄 오탈자 목록, 2010년 8월 8일

| | | | | |
|-----|----|-----|-------------------|---|
| | 14 | 278 | 밑에서 3번째 줄 | 예측률을 높다 → 예측률을 높인다 |
| ★★ | 15 | 282 | 소스 15-1 설명 | 메모리 의존성 ... r1 값이 같다면 → 메모리 의존성 ... r1의 값과 r7의 값이 같다면 |
| ★★★ | 15 | 282 | 각주 | ... 1, 2, 3 명령어가 먼저 수행될 수 있다 → 1, 2, 3 명령어 보다 먼저 실행될 수 있다 |
| | 15 | 285 | 아래에서 7번째 줄 | [조사탈락] 그런데 0x110 로드는 |
| | 15 | 288 | 아래에서 7번째 줄 | 변수 x,y,z,b는 → 변수 x,y,a,b는 |
| | 16 | 311 | 그림 16-6 | [부연설명] 2차원 배열을 접근 할 때, 특히 그림 (a)처럼 같은 행에서 인접한 열로 접근하는 것이 아니라, 그림 (b)처럼 행 단위로 접근할 때, 만약 한 행의 크기가 캐시 라인보다 크다면, 매번 캐시 미스가 일어날 수 있음을 뜻합니다. |
| | 16 | 311 | 소스 16-11 | 1: void MatrixMultiply(...) 5: a[i][j] += b[i][k] * c[k][j]; |
| ★★ | 16 | 315 | 4번째 줄 | 유효한 가상 메모리 주소가 같으면 → 유효한 가상 메모리 주소 값 같으면 (likely) |
| | 17 | 328 | 마지막 줄 | 2번 라인의 'sp=\$0.1'을 보면 → 2번 라인의 'sp=\$r0.1'을 보면 |
| | 17 | 329 | 각주 밑에서 두번째 줄 | VEX 컴파일러는 \$r10을 → VEX 컴파일러는 \$10.0을 |
| | 17 | 331 | 첫번째 문단 | \$0.3, \$0.4, \$0.5 → \$r0.3, \$r0.4, \$r0.5 |
| | 18 | 338 | 소스 18-4 | 9: lock 변수는 → lock_var 변수는 |
| ★★ | 18 | 346 | 소스 18-6 | "10: i < td->my_row_end"에서 <가 아니라 <=가 옳습니다. |
| ★★ | 18 | 347 | 소스 18-6 설명 | 열 단위로 → 행 단위로 |
| | 18 | 351 | 두번째 문단 4번째 줄 | 쉽지 않기 문제다 → 쉽지 않은 문제다 |
| ★ | 18 | 354 | 소스 18-3 5번째 줄 | [부연설명] for가 아닌 _for로 표기한 것은 Ct라는 언어에서는 컴파일러에게 힌트를 주고자 별도로 정의된 for와 유사한 매크로를 씁니다. 그러한 의미에서 _for로 한 것입니다. |
| | 18 | 355 | 3번째 줄 | 버전을 실지 않는다 → 버전을 심지 않는다 |
| ★ | 19 | 360 | 아래에서 4번째 줄 | Story 18의 <u>소스 18-3</u> → Story 18의 <u>소스 18-4</u> |
| ★ | 19 | 371 | 마지막 문단 첫번째 줄 | 그림 19-5 역시 → 그림 19-6 역시 |
| ★ | 19 | 372 | 아래에서 3번째 줄 | 그림 19-5는 → 그림 19-6은 |
| ★ | 19 | 374 | 두번째 문단 4번째 줄 | 그림 19-5가 → 그림 19-6이 |
| ★ | 20 | 378 | 그림 20-1 밑 | [부연설명] 1번 프로세서가 → 1번 프로세서(TestThread2)가 |
| | 20 | 380 | 소스 20-2 | CACHE_ALIGN_VC → CACHE_ALIGN_MSVC |
| ★★★ | 20 | 384 | 두번째 문단 아래 둘째 줄 | t1:3 -> t2:3 -> <u>t1:3</u> -> t2:4 → t1:3 -> t2:3 -> <u>t1:4</u> -> t2:4 |
| ★★★ | 20 | 385 | 두번째 문단 두번째 줄 | 그렇다 하더라도 <u>락이</u> → 그렇다 하더라도 <u>IMOI</u> |
| | 20 | 385 | 아래에서 두번째 줄 | __atmoic → __atomic |
| | 20 | 386 | 첫번째 문단 아래 둘째 줄 | TM 구현 문제뿐만 특히 → TM 구현 문제뿐만 아니라 특히 |