

likely, unlikely 함수란?



moreNice · IT 개발/리눅스 커널
2012.01.06 00:04



리눅스 커널코드를 보면, likely와 unlikely란 함수를 종종 발견하곤 한다. 컴파일러는 작성된 함수와 코드들에 대해서 최적화 작업을 하여 성능을 향상시킨다. 여기서 갑자기 컴파일러 이야기를 하는 이유는 해당 함수들(정확히 말하자면 매크로이긴 합니다만)은 컴파일러와 관련이 있다.

likely와 unlikely는 if 분기문에서 likely를 주면 컴파일러에게 해당 분기문에서 참인 경우가 더 많을 것이라는 정보를 주어 성능을 추가적으로 향상시키는 함수이다. 반대로 if 분기문 안에 unlikely를 주면 거짓인 경우가 더 많을 것이라는 정보를 준다.

실제로 정의된 매크로를 따라가면 `__builtin_expect`라는 함수가 사용됨을 볼 수 있다. `__builtin_expect`라는 이름이 좀 더 실제 의미에 적합한걸로 보이지만 likely라는 이름이 센스 있어 보인다.

```
#define likely(x) __builtin_expect((x),1)
#define unlikely(x) __builtin_expect((x),0)
```

likely와 unlikely는 커널에서 define된 매크로라서 사용하면 컴파일 오류가 발생한다. `__builtin_expect`를 사용하면 일반 C 어플리케이션에서 사용할 수 있다.

분기문에서 명백히 참이나 거짓 중 하나가 명백히 자주 사용될 것으로 판단되어 진다면 이 함수를 활용할 경우 성능을 높일 수 있을 것이다. 개인적으로 비용이 많이 드는 데이터처리에서 사용할 때 도움이 될게 같다. 마지막으로 참이나 거짓이 어떻게 나올지 어떤 것이 자주 나올지 확실하지 않을 경우에는 사용하지 않고 컴파일러가 알아서 하게 하는게 좋을 것이다.

이해를 조금 더 돕기 위한 테스트용 예제를 만들어보았다. 테스트용으로 급조해서 만든 것이라 현실감은 떨어지니 이해바란다.



```
#include <stdio.h>
```

```
int test(int i)
{
    if( __builtin_expect(i > 0, 1) )
    {
        i += 4;
        i *= 2;
    }
    else
    {
        i -= 4;
        i /= 2;
    }

    return i;
}
```

```
int main()
{
    printf( "%d\n", test(3));
    return 0;
}
```



좋아요 친구들이 무엇을 좋아하는지 알아보려면 가입하기



방명록 검색 로그아웃



크리에이티브 커먼즈 라이선스



이 저작물은 [크리에이티브 커먼즈 코리아 저작자표시-비영리-동일조건변경허락 2.0 대한민국 라이선스](#)에 따라 이용할 수 있습니다.



Tags

[kernel](#) [likely](#) [linux](#) [unlikely](#)

Read Next



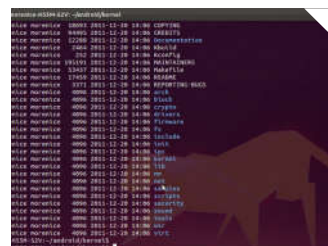
ubuntu 데스크탑과 서버 커널의 차이점은?

2013.01.08



어떻게 Linux kernel 공부를 시작하죠?

2013.01.01



안드로이드 4.0, 기본 커널과 삼성 커널 코드 비교

2011.12.28



안드로이드 커널 버전과 플랫폼 버전 확인하는 방법

2011.12.27

Comment 0



여러분의 소중한 의견을 들려주세요.



[방명록](#) [검색](#) [로그아웃](#)



이름

비밀번호



ADD COMMENT

