Apache, Mysql, PHP 입문하기

이홍일*skycry@kldp.org* <mailto:skycry@kldp.org>

v, 2000년 01월부터 05월까지

sgml 변환: 고은정*GOWILL@chollian.net* <mailto:GOWILL@chollian.net>

2000년 9월 20일

- IRC_NIC : skycry, matus, cry
- Startcraft_ID : skycry, -=skycry=-
- Quake_ID : skycry
- 포트리스_ID : skycry
- 소메_ID : skycry
- 이하 모든 통신회사 Mail_ID : skycry
- 앞으로의 계획 : 메신저 소켓 만들기와 웹메일 연동 만들기, 그리고, 모두 공개하기

주의.

지금부터 시작할 모든 내용은 저자의 나름대로 생각이며, 이로 인한 피해는 책임지지 않습니다. (단, 글의 의문점은 메일로 문의하기 바랍니다.)

들어가기 앞서

글의 내용은 처음 입문하는 초보자를 위주로 저가 처음 접했을 때부터 아무것도 모르는 상태에서 집필한 글이며.. 다소 무식 하고 무지막지한 방법도 있을 것이며 잘 알려진 방법도 있을 것입니다. 단지 이 글은 아무것도 모르는 이가 처음 입문을 할 경우 도움을 주기 위해 쓰여 졌습니다.

참고로⋯.

이 글은 linux에 입문한지 얼마 되지 안고, 웹프로그래밍에 관심이 있으며 공개된 모든 irc서버에서 고수를 찾아 돌아 다녔지만 통 대답을 듣지 못한 이, 또한 여타 잘 알려진 linux모임 게시판에 질문 했을 때 linux는 유어셀프니 머시기니라는 말만 들은 사람⋯ 여타 초보자의 설움을 맨날 느끼는 이 그리고, 모출판사 실장님과 신촌 로타리에 위치한 한 서점 사장님, 마지막으로 아들 잘되는 날만 기다리시는 울 엄니에게 바칩니다.

목차

제 1 절 웹 서버와 전자상거래의 대략적인 설명

PHP를 접하기에 앞서 이 장은 웹 서버의 역할과 요즘 한창 이슈가 되고 있는 전자상거래와 쇼핑 몰에 관해 간단히 알아 보겠습니다.

그림 1: 그림. 1-1 웹 서버와 클라이언트

1.1 웹 서버의 역할

월드와이드웹은 인터넷을 보다 더 쉽게 항해 하게끔 발전되어 영화나 책을 보듯이 여러분윈앨앖앖쐬都鳴×痼笭읍求墨麻 앞으로도 더 사용자 중심으로 발전할 것입니다. 우리가 이렇게 안방에서 웹 브라우저로 볼 수 있는 것도 웹 서버가 있어야지만 가능한 것입니다. 웹을 이용한 인터넷의 대중적인 확산이 커짐에 따라 웹 서버의 기능도 다양해져 갔습니다. 하지만 옛날의 웹 서버의 역할은 파일의 송수신, 텍스트의 링크, 게시판 정도 였습니다. 지금은 대화방에서부터 상품의 쇼핑, 오락, 전화, 경매에 이르기까지 그 사용 용도가 다양해졌습니다. 우리가 일상에서 하던 모든 일을 웹상에서 해결하는 상태가 된 것입니다. 이 중에도 웹상에서 쇼핑을 하고 선택한 물건을 주문하는 쇼핑 몰 사이트는 요즘 들어 한창 부각 중입니다. 안방에서 책과 음악 앨범, 옷 등 원하는 것들을 구입할 수 있고 게다가 전자 결재도 가능해진 것입니다. 이렇듯 웹 서버의 역할이 점점 더 커지기 때문에 서버의 운영자들은 보다 강력하고 안정적인 웹 서버를 선택하게 됩니다. 웹 서버의 역할을 간략히 그림으로 확인하면 다음과 같습니다.

1.2 웹 서버의 종류

대중의 인기를 끌었던 웹 서버의 종류는 Apache , IIS, Netscape, NCSA, WebSTAR , CERN, MacHTTP로 요약할 수 있습니다.

- Apache - 유닉스 기반의 웹 서버로 발전 했으며 마이크로소프트 윈도우 환경도 지원합니다. 현존 하

는 서버 중 가장 많은 점유율을 가지고 있습니다.

- IIS(Internet Information Server) - NT 기반의 웹 서버이며 주로 사용되는 스크립트 언어는 ASP입니다. 하지만 최근에는 PHP와의 접목도 이루어지고 있습니다.
- NCSA - 유닉스 기반이고 아파치 웹 서버의 초기 모델이며 현재는 아파치 웹 서버왼앨얹얹쐬纛美®를 물려준 상태입니다.
- Netscape, CERN - 유닉스 기반이며 현재 Apache에 밀려 사용자가 현격히 감소한 상태 입니다. 현재 각 웹 서버의 사용자 층을 확인할 때 Apache 웹서버가 나머지 웹 서버의 사용자를 모두 합한 것을 넘어선 상태입니다.

1.3 웹 서비스의 종류

웹 서비스의 종류는 다양한 사용자의 욕구와 이를 만족 시키기 위한 개발이 지속적으로 이루어지는 한 계속해서 증가 할 것 입니다. 물론 같은 서비스라도 경쟁을 위해 차별화를 추구할 것입니다. 웹 서비스의 예를 보면 여러 가지 콘텐츠를 이용한 다중 쇼핑 몰(전자 제품, 가정 용품, 의류, 잡화)에서 한가지 상품을 특화한(농산물, MP3, 공연 예매, 스포츠 용품···)사이트 각 개인의 정보와 여러 가지 디렉토리 서비스, 일반적인 자료의 공유와 정보 교환 사이트 등 여러 방식이 존재 합니다. 물론 이것은 차별화를 위한 하나의 방편 입니다. 차별화를 위해서는 디자인도 중요하지만 내용이 우선해야 됩니다. 좋은 예로 뉴욕 타임즈를 보면 알 수 있습니다. 즉, 인지도에 의한 해당 쇼핑 몰의 인기도가 중요한 역할을 합니다. 이를 극복하는 것은 오직 차별화 입니다.

다중 쇼핑몰의 예

특화 쇼핑 몰의 예

1.4 전자 상거래

가장 포괄적으로 생각하는 전자 상거래의 의미는 인터넷을 이용하여 돈에 해당하는 물건을 주고 받는 것입니다. 돈의 가치를 지닌 전자화폐를 주고 원하는 물건을 살 수도 있습니다. 다시 말해 네트워크를 통한 상품의 거래이며 광고, 기타 모든 서비스의 구매 활동 등을 말합니다. 이것은 실물 경제와는 다른 또 다른 경제 형태가 될 것입니다. 하지만 아직 이러한 전자 화폐의 지불과 물건 구입은 이루어지지 않습니다. 다만 사이버 머니 개념의 소액전자화폐 등을 이용한 이벤트형식의 물건 거렵뜬뜬뜬듯÷肩獰沮側資筝읍求墨痲

1.5 전자 상거래의 문제점

신용카드를 이용하거나 은행 지로 번호를 통한 돈의 거래는 활발히 이루어 지고 있지만 정작 전자 화폐의 이용은 아직 보안과 여러 다른 여러가지 이유로 인해 개발이 늦어 지고 있습니다. 만약 이러한 여타의 문제들이 해결 된다면 인터넷 쇼핑몰은 한층 인기가 상승할 것 입니다. 전자 화폐의 근본적인 문제는 물론 돈의 발행, 이중성 , 통화 , 복제 등 여러가지가 있을 수 있습니다. 이러한 문제점이 해결되고 안정적인 전자 결재가 이루어 진다면 은행의 역할 역시 감소할 것입니다. 이것은 각 은행의 수수료 수입의 감소와 자금 조달 방법 또한 새로워 지게 하기 때문 입니다. 세계적으로 이러한 전자 상거래의 표준화 작업은 각국의 독특한 방식으로 진행 중이며, 시장을 선점하기 위해 치열한 노력을 하고 있습니다.

전자 상거래 관련 사이트

전자상거래 위원회 - *http://www.ecc.or.kr/* `<http://www.ecc.or.kr/>`

<TIP>——————————————————————————-

전자상거래 자격증

그림 2: 그림. 1-3 다중 쇼핑몰

그림 3: 그림. 1-4 농산물 쇼핑몰

그림 4: 그림. 1-5 전자상거래 위원회

그림 5: 그림. 1-6 전자상거래 자격증 관련 과목

전자상거래 위원회에서는 전자상거래 관리사 자격증을 신설했으며 이로 보다 전문적으로 사이트를 구성하고 주도하는 인력을 배출할 것입니다. 보다 자세한 내용은 해당 사이트를 방문하면 정보를 얻을 수 있습니다. *http://www.ecc.or.kr/* `<http://www.ecc.or.kr/>`

관련 과목

——————————————————————————————————————-

- 정보통신부 - *http://www.mic.go.kr/* `<http://www.mic.go.kr/>`
- 한국무역 정보통신 - *http://www.ecrc.or.kr/* `<http://www.ecrc.or.kr/>`
- 한국전자상거래 관리사 협회 - *http://www.ecca.co.kr/* `<http://www.ecca.co.kr/>`
- 한국 전자거래 협의회 - *http://www.kcals.or.kr/* `<http://www.kcals.or.kr/>`
- 중소기업진흥공단 전자상거래 지원 센터 - *http://ecrc.smipc.or.kr/* `<http://ecrc.smipc.or.kr/>`
- 한국 생산성 본부 전자상거래 지원 센터 - *http://ecrc.kpc.or.kr/* `<http://ecrc.kpc.or.kr/>`
- 한국유통 정보 센터 - *http://www.eankorea.or.kr/* `<http://www.eankorea.or.kr/>`
- 한국 전자거래 진흥원 - *http://www.keb.or.kr/* `<http://www.keb.or.kr/>`
- 한국 표준 협회 - *http://www.kisi.or.kr/* `<http://www.kisi.or.kr/>`
- 한국 건설기술 연구원 - *http://www.kisi.or.kr/* `<http://www.kisi.or.kr/>`

- 한국 전자상거래 관리사 협의회 - *http://www.ecca.co.kr/* `<http://www.ecca.co.kr/>`
- 한국 커머스넷 - *http://www.commercenet.or.kr/* `<http://www.commercenet.or.kr/>`
- 정부EDI/EC 지원 센터 - *http://ediweb.nca.or.kr/* `<http://ediweb.nca.or.kr/>`

1.6 전자 상거래 동향과 개선할 부분

한국 전산원의 통계에 따르면 1999년 기준으로 국내 쇼핑 몰 사이트의 수는 400개에 육박하며 현재 계획 중이거나 활동중인 사이 계속 증가하는 추세 입니다. 각 쇼핑 몰의 경우 인지도에 따라 서비스 콘텐츠에 따라 활동이 활발한 사이트는 일부 입니다. 대표적으로 유니플라자 , 삼성인터넷 쇼핑몰 , 롯데 백화점을 들 수 있습니다. 국내의 전자 상거래 규모는 600억 정도의 규모로 예상 하며 본격적인 전자 상거렵똔뜬뜬듯÷肩獰沮嗤浴誰맛퓰첾揚뻐kV霞寗痴萬〉뎬僬柰痼막觀맛都求墨庣 이러한 수준에 도달 하기 위해서는 표준화가 이루어 져야 되고 보안 , 유통채널의 다변화 등 여러 가지 개선할 부분들이 존재합니다. 무엇보다 수익성 있는 콘텐츠 개발 또한 중요한 부분 입니다.

1.7 마무리

이 장에서는 PHP를 접하기 전에 웹서버 종류에 대해서 간략히 살펴 보았습니다. 현재 가장 각광 받는 쇼핑 몰과 이를 위한 전자 상거래에 관한 것을 보았습니다. 다음 장에서는 웹서버 구축에 필요한 Apache , Php , Mysql를 알아보겠습니다.

제 2 절 웹서버 구축에 필요한 것들 (PHP , APACHE , MYSQL)

2.1 Apache , Php , Mysql 소개

웹 개발에는 많은 도구가 필요합니다. 하드웨어를 비롯하여 소프트웨어에서도 그 종류는 많습니다. 크게 웹서버프로그램, 데이터베이스 그리고, 개발 언어입니다. 모든 운영체제에서 웹서버를 구축할 수 있고 또한 공개 프로그램이면서도 기능이 뛰어난 아파치 웹서버와 Mysql 데이터베이스, PHP 언어에 관하여 소개를 하겠습니다.

2.1.1 Apache 웹 서버

아파치 웹 서버는 대중의 힘을 바탕으로 가장 큰 인지도를 얻을 만큼 사용자가 이끌어 가는 서버 입니다. 여타의 상용 서버를 밀킁칠캳캳쳉牆 우뚝 설 수 있었던 이유는 오픈 소스 개발에 따른 보다 빠른 사용자 중심의 개발로 인한 수많은 개발자의 합류 입니다. 무엇 보다 상용 서버에 뒤지지 않는 신속한 버그 PATCH와 끊임없는 수많은 개발자의 지원과 안정성 입니다. 이러한 점으로 웹 서버 시장에서 다른 서버와 비교해 웹 서버 시장에서 과반수 이상의 점유율을 보이고 있습니다.

공식 아파치 사이트 외에 각 국가별 그룹별로 사이트가 존재 하는데 한국 아파치 그룹에 가면 한글화 된 아파치 서버의 최신 정보와 기술적인 문제를 볼 수 있습니다.

Apache 웹 서버의 역사 아파치 프로젝트의 시작은 처음 1995년경 대중의 사랑을 받던 NCSA의 개발자 중 일부가 모여 시작 했습니다. 이것을 시발로 NCSA HTTPD 개발자와 아파치 개발자들이 합류되고 이전에 만들어졌던 아파치 0.6.2를 완전히 개선한 0.8.8을, 그리고 아파치 1.0을 1995년10월에 만들어 냈습니다. 아파치 서버는 단기간에 기존의 NCSA 자리를 밀킁칠캳캳쳉茨골招×철享읍求墨庣 이 상승세는 지금도 계속 중이며 다른 여타의 서버가 넘보지 못하는 자리에 있게 되었습니다. 이것은 물론 직접 관여하는 개발자 이외에 전세계 아파치 그룹 및 사용자들이 항상 테스트 하고 패치하는 작업의 결과입니다. 다른 서버가 넘보지 못하는 부분이 이런 오픈 소스 개발일 것입니다. 아파치는 멈추지 않는 것입니다. 누군가 멈추려 해도 절대 멈추지 않는 지속적인 개발이 이루어 지는 것입니다.

그림 6: 그림. 2-1 Apache 공식 사이트(*http://www.apache.org* `<http://www.apache.org>`)

그림 7: 그림. 2-2 한국 Apache 그룹(*http://www.apache.kr.net* `<http://www.apache.kr.net>`)

그림 8: 그림. 2-3 코만치 사이트(*http://www.comanche.org* <`http://www.comanche.org`>)

Apache GUI Tool 아파치를 사용자 중심의 웹 서버로 만드는 노력도 진행 중입니다. GUI 프로젝트 즉, 코만치 프로젝트라 불리는 이 작업은 아파치의 운용을 그래픽적으로 할 수 있게 하는 것입니다. 많은 부분이 완성 되었고 리눅스 배포판에도 포함되어 있습니다. 아파치 서버는 모든 설정이 텍스트 파일인 httpd.conf를 수정하는 방법이기 때문에 다소 그래픽 환경에 익숙한 사람왼앨얹얹쐬都쟌盧灼寗層돚斂雌∮謀歐瘦箏慕恝咀觀隙缺殮粹kt®에 이 프로젝트는 앞으로 아파치서버의 발전에 중요한 부분을 차지하게 될 것입니다.

코만치 사이트에서 코만치에 관한 정보와 각 환경에 따른 코만치 설치 파일을 다운 받을 수 있습니다.

각 환경에 따른 지원 파일

- w Linux binary
- w Windows 9x/NT/2000 binary
- w HP-UX PARISC binary
- w Solaris Sparc binary
- w Irix MIPS binary

코만치 메인화면을 보면 기존의 httpd.conf에서의 설정을 정말 간단히 깔끔하게 수정 할 수 있게 개발 된 것을 볼 수 있습니다. 이것은 기존의 텍스트 환경에서 작업 하던 사람 또한 쉽게 접근하게 되어 있습니다.

그림 9: 그림. 2-4 코만치 메인 화면

그림 10: 그림. 2-4 코만치 메인 화면

코만치의 Properties configuration 화면

위 그림들을 보면 기존의 설정하는 부분을 아주 쉽게 생성하고 수정할 수 있게 되어 있음을 알 수 있습니다. 코만치는 또한 여러 운영체제별로 실행 파일을 제공하고 있으며 계속 개발 중입니다. 물론 아직 개선할 점을 찾으며 계속 개발 중인 윈도우용 아파치를 지원하는 코만치 또한 존재합니다.

2.1.2 Apache 사용자 현황

전세계 웹 서버 시장을 볼 때 아파치 서버의 사용은 월등합니다. 이것은 무엇보다 공짜임에도 불구하고 강력한 기능과 빠른 패치 따른 이유입니다. (아파치 서버의 사용자 현황은 www.netcraft.com의 자료를 인용 했습니다.)

netcraft 웹 서버관련 통계

위 그림에서 아파큥칠캭캭쳉⟨摸IJBIJŋ버 사용자 수를 모두 합한 것 보다 높은 것을 볼 수 있습니다.

위 그림들에서 볼 수 있듯이 아파치 서버는 발전면에서 그리고 최고 사용 서버에서 모두 월등히 높음을 알 수 있습니다.

위 그림은 최근 몇 년간 각 서버의 변화 모습입니다. 2000년 현재 아파치 서버를 가동 중인 도메인은 400만개 이상이 되는 걸 볼 수 있습니다.

그림 11: 그림. 2-5 Properties configuration1

그림 12: 그림. 2-6 Properties configuration2

그림 13: 그림. 2-7 Edit Element

그림 14: 그림. 2-8 윈도우용 코만치

그림 15: 그림. 2-9 모든 도메인의 사용중인 서버 현황

그림 16: 그림. 2-10 Top Developers

그림 17: 그림. 2-11 Top Servers

그림 18: 그림. 2-12 몇 년간 웹 서버 사용 변화

아파치에 관한 문서를 잠시 보면 웹 프로토콜의 독점을 막아야 된다고 말하고 있습니다. 인터넷에서 특정 기업에 의한 독점이 이루어 진다면 그것은 인터넷의 자유로움과 자유 경쟁을 위배하는 것입니다. 이러한 이유로 아파치는오픈 소스 프로젝트를 지향하고, 무료로 배포하는 이유 중에 하나입니다. 이 시간에도 전세계 수많은 개발자와 사용자들이 아파치를 보면서 문제점이 있는지 쩽죷죽죽잣改構資聆甖痼都求墨麻 자신이 아파치 서버에 만족하고 기능 개선을 바란다면 뉴스 그룹에 어떠한 글을 올려도 도움이 될 것입니다. 이런 응답 글 하나하나가 아파치 서버 사용자와 개발자들왼앨얹얹쐬讀傷鄂祁동遲絹É 것이며 이렇게 발전한 이유입니다.

2.2 PHP

2.2.1 PHP 소개

PHP는 처음 웹 스크립트 언어를 접하는 이왼앨얹얹쐬牘릿毋Э痢庸 보다 빠른 시간에 웹을 다이나믹 하게 만들어 줍니다. 이것은 빨리 변화되는 인터넷 시장에서 중요한 역할을 하는 것입니다. 이러한 PHP(Professional HyperText Preprocessor)는 서버에서 해석되는 HTML에 내장되어 동작하는 스크립트 언어입니다. C, Java, Perl 등에서 많은 문장 형식을 빌려왔으며 웹 브라우저 등으로 실제 코드를 볼 수 없다는 것에 보안상 유리한 점도 있습니다. 무엇보다 이 언어는 웹을 개발하는 이왼앨얹얹쐬牘릿茅磁®한 환경을 제공해줍니다. 또한 CGI 프로그래밍 언어(C, Perl)와 ASP들이 만들어 냈던 BBS, 웹메일, 쇼핑 몰에 이르기까지 대부분을 쉽게 만들 수 있으며 상업, 비상업적으로 얼마든지 사용 가능한 언어입니다. 설치 환경은 Linux, Unix, 윈도우 등 대부분의 운영체제를 지원합니다.

2.2.2 PHP가 지원하는 데이터베이스

현존하는 데이터베이스를 거의 지원합니다. 간단한 게시판이나 방명록을 만들 경우라도 요즘은 데이터베이스를 사용합니다. 무엇보다 빠르게 검색 가능하면서 체계적인 저장이 되므로 데이터 관리가 편리하다는 이유입니다. 한 두개의 게시물일 경우와 수백 수천개의 게시물이 있을 때, 후자의 경우라면 보다 편리한 사용 방법은 데이터베이스 구축입니다. 지원하는 DataBASE는 (Adabas D, Dbase, Oracle, Empress, PostgreSQL, FilePRO, Solid, Informix, Sybase, InterBASE, Velocis, Msql, Unix dbm)등이 있으며, PHP4 ZEND 의 경우는 더 확실한 지원을 보장 할 것으로 보입니다. 앞으로도 데이터베이스와의 쉬운 연결과 이식성 등으로 애용자가 날로 증가하리라 봅니다.

2.2.3 PHP의 역사

PHP는 1994년 가을 Rasmus Lerdorf에 의해 만들어졌으며, 그의 홈페이지에서 공개되어 사용 되기는 1995년 초 Persnol Home Page Tool라고 불려지면서 입니다. 이것은 몇 개의 매크로를 가진 파서(명령 해석) 엔진과 홈페이지 뒤에서 움직이는 몇 개의 유틸리티로 이루어진 것이었습니다. 또 다시 95년 중반 이 파서가 재작성 되어 PHP/FI Version 2라 불리어 졌습니다. 또, FI는 Rasmus가 작성한 HTML 형식의 데이터를 해석할 수 있는 별도의 패키지 였는데, 그는 이 둘을 합쳐 Msql을 지원하도록 하여 PHP/FI를 만들어 냈습니다. PHP/FI는 많은 사람들의 공헌에 힘입어 발전을 거듭했으며 1996년 후반 전세계적으로 15,000개 이상의 웹사이트에서 사용되어졌고 1997년 중반엔 50,000정도 이상으로 늘어났습니다. 그리고 이해 PHP/FI는 Rasmus 개인이 아닌 팀에 의해 개발되어졌습니다. Zeev Suraski와 Andi Gutmans에 의해 재작성되어 PHP 3 가 만들어졌고, 이것은 이전 버전에서 많은 기능들을 옮겨왔으며 새로운 기능들을 추가했습니다. 정확한 통계자료는 아니지만 NETCRAFT의 통계 자료를 보면, 현재 전세계 300,000대 이상의 웹 서버에서 사용중입니다. 만약 도메인에서 사용중인 PHP를 보면 두배 이상이 될 것입니다. 앞으로도 아파치 웹서버의 증가와 PHP4 ZEND의 다양한 포맷 지원으로 더 증가하리라 생각합니다. (PHP3의 FAQ와 Andi Gutmans의 글을 참조 했습니다)

그림 19: 그림. 2-13 PHP 공식 사이트(*http://www.php.net/* `<http://www.php.net/>`)

2.2.4 PHP3와 PHP4 ZEND

PHP3와 PHP4를 따로 배울 필요는 없습니다. 다시말해 이 둘은 프로그램의 상위 버전이라고 생각하면 될 것입니다. (PHP4 앞으로 ZEND라함) ZEND는 PHP3가 보다 더 많은 기능을 넣기위해 복잡해진 스크립트로 인해 성능이 떨어지는 것을 본 Andi Gutmans와 Zeev Suraski에 의해 보다 원활하게 동작하게 만든 새로운 엔진 입니다. 또한 ZEND 엔진은 PHP 이외의 다른 프로그램에서도 재사용될 수 있고 앞으로 다른 언어에서도 사용 가능하리라 봅니다. ZEND 엔진을 사용하면 PHP3보다 300배 정도 성능 향상을 가져오며, 사용자가 증가했을 때 확연히 보입니다. 여기에 PLUG-IN을 별도로 구매하면 더 향상을 가져올 것입니다. 그리고 ZEND는 foreach, class::method() call과 같은 새로운 문법과 PHP3에서의 사용하기에 불편했던 점을 상당히 개선 했습니다. 강화된 것을 보면, 우선 이전 버전보다 더 많이 모듈화 되었습니다. PHP3에서와는 달리 동적인 모듈화가 쉽게 되도록 설계 되어진 것입니다. 그리고 표준 CGI, Apache, Fhttpd, ISAPI(MS의 IIS를 위한), NSAPI(Netscape server을 위한)를 지원하게 되었습니다. 뿐만 아니라 기존의 ASP를 통해서 이용한 COM/DCOM까지 ZEND는 영역을 넓혔습니다. 현재 PHP4 ZEND는 candidate 1까지 나왔으며 앞으로 계속해서 업그레이드 될 예정입니다. ZEND Optimizer의 경우 현재 beta 2까지 나와 있으며 Free Down이 가능 합니다. 설치 환경은 PHP4 beta4 pl1과 이후 버전(PHP4 RC1)에서 설치 가능하며 각각의 시스템 환경을 보면 아래와 같습니다.

- x86 Linux glibc2.1-based systems (tested on Red Hat 6.1, Mandrake 7.0, Slackware 7.0 and SuSE 6.2).
- x86 Linux glibc2.0-based systems (tested on Red Hat 5.2 and SuSE 6.1)
- x86 Linux libc5-based systems (tested on Slackware 4.0 and Debian 1.3.1r8)
- Sparc Solaris 7
- FreeBSD 3.4 and 4.0
- Windows NT 4.0 (not tested on other versions of Windows)

www.zend.com으로 가면 간단한 가입 절차와 함께 메일로 패스워드를 받고 로그인 후 ZEND Optimizer를 다운로드 할 수 있습니다.

위 그림에서 zend , 웹 서버 , 데이터베이스 , 브라우저의 역할에 대한 간략한 구조를 확인 할 수 있으며 zend 엔진이 이러한 구조에서 담당하는 부분을 확인 할 수 있습니다.

2.2.5 PHP와 ASP

PHP와 ASP는 근본적으로 서버용 객체 지향적인 스크립트 언어라는 점에서 같지만, ASP의 경우 IIS, PWS와 같은 윈도우 환경에서 움직이는 서버를 지원하는데 반해 PHP는 Apache, IIS, PWS, 서버 등과 유닉스 윈도우 환경에서 움직이는 모든 서버를 지원합니다. 또한 PHP는 위에서 언급한 데이터베이스 같은 경우 ODBC에 의한 연결과 유닉스에서 사용되는 거의 모든 DB를 연결해 줍니다. 무엇보다 매력적인 것은 PHP 코드 내에서 사용자 자신의 객체를 정의해서 사용할 수 있다는 점입니다. 이것 말고도 여러 가지 장점들이 많이 있습니다.

2.3 Mysql

2.3.1 Mysql에 관해서

Mysql 데이터베이스는 다른 상용 데이터베이스와 비교해 안정성과 기타 여러 문제에 크게 떨어지지 않으면서 공개되어진 데이터베이스입니다. 다양한 함수와 여러 기능면에서 상용 데이터베이스가 가지지 못한 점을 가진 것도 많이 있습니다. 그리고 공개된 다른 데이터베이스와 비교 할 때 사용자 또한 넓습니다. Mysql은 C , C++, Java, python, tck/tk, ODBC도 제공하며 다양한 웹서버와의 연결도 용이합니다. ODBC는 Ms 윈도우(win95 , win98 , NT) 등을 제공합니다. 현존하는 윈도우의 대부분을 지원 하며 설치

그림 20: 그림. 2-14 ZEND 공식 사이트(*http://www.zend.com* `<http://www.zend.com/>`)

그림 21: 그림. 2-15 zend Optimizer

그림 22: 그림. 2-16 zend 엔진의 역할

그림 23: 그림. 2-17 Mysql 공식 사이트(*http://www.tcx.se/* `<http://www.tcx.se/>`)

또한 대부분의 OS를 지원 합니다. Mysql을 개인적인 용도로 사용할 경우 무료이며 상업적인 목적으로 사용할 경우 mysql 개발팀에 조금의 후원을 하면 가능합니다.

Mysql이 지원하는 OS와 패키지

- AIX 4.x with native threads
- BSDI 2.x with the included MIT-pthreads package
- BSDI 3.0, 3.1 and 4.x with native threads
- DEC UNIX 4.x with native threads
- FreeBSD 2.x with the included MIT-pthreads package
- FreeBSD 3.x with native threads
- HP-UX 10.20 with the included MIT-pthreads package
- HP-UX 11.x with the native threads.
- Linux 2.0+ with LinuxThreads 0.7.1 or glibc 2.0.7

- MacOS X Server
- NetBSD 1.3/1.4 Intel and NetBSD 1.3 Alpha (Requires GNU make)
- OpenBSD > 2.5 with native therads. OpenBSD < 2.5 with the included MIT-pthreads package
- OS/2 Warp 3, FixPack 29 and OS/2 Warp 4, FixPack 4
- SGI Irix 6.x with native threads
- Solaris 2.5, 2.6 and 2.7 with native threads on SPARC and x86
- SunOS 4.x with the included MIT-pthreads package
- SCO OpenServer with a recent port of the FSU Pthreads package
- SCO UnixWare 7.0.1
- Tru64 Unix
- Win95, Win98 and NT

2.3.2 Mysql의 주요 기능

Mysql은 C , C++ 로 작성 되어졌으며 많은 테스트를 통해서 안정성을 입증했습니다. 또한 SQL문의 대부분을 완벽하게 지원하며 각 테이블에 16개의 인덱스를 생성 할 수 있습니다. 큰 규모의 테이블을 생성할 수 있으며 원격으로도 접속 가능 합니다. 다양한 컬럼 타입을 제공하며 사용자 관리를 쉽게 할 수 있습니다. 이외 다양한 기능들이 있으며 모든 스크립트들이 마찬가지 이지만 데이터베이스와의 유연한 연동은 필수가 되었습니다. Mysql 역시 이점이 탁월하며 모든 플랫폼에서 지원되므로 사용 범위 또한 넓은 것입니다. 처음 mysql이 Tcx 내부에 사용될 목적으로 만들어 졌습니다. 이것은 그 당시의 여타 SQL서버 보다 성능이 좋아야 했습니다. 이처럼 뛰어난 데이터베이스를 현재 우리가 무료로 사용하고 있는 것입니다.

2.3.3 Mysql GUI Tool

Mysql 역시 일반 텍스트 기반이 아닌 그래픽적으로 쉽게 작업할 수 있는 GUI Tool이 있습니다. 이 Tool의 기능들은 물론 스크립트로 만들 수도 있습니다. 이렇게 대부분의 프로그램들은 처음 텍스트 기반하지 않고 사용자 중심의 GUI 기반의 Tool을 생성함으로써 보다 넓은 사용자 층을 확보하고 있습니다.

GUI Tool은 유닉스 기반에서 처음 만들어 졌습니다. 지금은 윈도우 기반의 툴 또한 다양한 기능을 제공하고 있고 데이터베이스의 원격관리에서 다양한 사용자 편의를 제공합니다. 일반적인 Telnet 접속을 해서 데이터베이스를 관리하는 방법에 식상한 사람들과 처음 데이터베이스를 접하는 사람들에게 좋은 방법을 제시합니다.

MS 윈도우 용 GUI Tool

다음 장에서는 리눅스 환경에서 Apache , Php , Mysql 설치를 자세히 알아 보겠습니다.

제 3 절 리눅스에서 APM 설치하기

APACHE , PHP , MYSQL을 Linux에서 설치 하는 방법을 알아 보겠습니다. 위 프로그램들은 최근 업그레이드 된 걸로 여러분들은 설치하세요. 최근 버전일수록 설치하는 과정은 조금 다를 수 있습니다. RPM 패키지로 인스톨 하는 방법도 있지만 설치할 때 사용 옵션의 추가와 위치 등 여러 가지 장점이 있는 TAR 패키지를 이용하여 설치하도록 하겠습니다.

A.P.M(Apache, Php, Mysql)의 설치는 다음 순서로 진행합니다.

그림 24: 그림. 2-18 Mysql GUI Tool(Linux)

그림 25: 그림. 2-19 윈도우용 Mysql GUI Tool 사이트(*http://www.winmysql.com/* `<http://www.winmysql.com/>`)

그림 26: 그림. 2-20 윈도우용 Mysql GUI Tool row 화면

그림 27: 그림. 2-21 윈도우용 Mysql GUI Tool 데이터베이스 접근 화면

그림 28: 그림. 2-22 윈도우용 Mysql GUI Tool query 화면

그림 29: 그림. 2-23 윈도우용 Mysql GUI Tool 데이터베이스 사용자 권한 설정 화면

그림 30: 그림. 2-21 윈도우용 Mysql GUI Tool 데이터베이스 접근 화면

3.1 A.P.M을 설치하기 전에 할 일

설치는 APACHE에 모듈을 추가해서 컴파일하겠습니다. 최근의 업그레이드된 프로그램은 아래의 사이트에서 구할 수 있습니다.

- Apache - *http://www.apache.org/* `<http://www.apache.org/>`
- Php - *http://www.php.net/* `<http://www.php.net/>`
- Mysql - *http://www.mysql.org/* `<http://www.mysql.org/>`

1)Apache, PHP, Mysql 패키지 옮기기

먼저 CD에 있는 APACHE, PHP, MYSQL을 임시 디렉토리에 옮겨 놓습니다. 여기서는 /TMP 디렉토리에 옮겨서 설치를 하겠습니다. 여러분의 임의 디렉토리로 옮겨서 사용하면 됩니다.

<TIP>——————————————————————-

CD 마운트하기

```
CD 마운트에 익숙치 못한 분을 위해 마운트 하는 방법을 간단히 설명하겠습니다.
/etc/fstab 파일의 내용중 cdrom 항목이 대부분 인스톨 시에 있기 때문에 다음과 같이 합니다.

  -.fstab 의 cdrom 항목이 auto 일 경우
        auto로 설정되어 있다면 부팅 전 cd가 있을 경우 /mnt/cdrom 디렉토리에 가면 cd 내용을 볼 수 있습니

  -.fstab의 cdrom 항목이 auto가 아닐 경우
        cd를 넣은 다음
        #mount /mnt/cdrom 하면 cd 내용을 볼 수 있습니다.

  -.또 다른 방법
        레드햇 6.0 버전 이라면 Xwindow에서 cdrom을 마운트와 언마운트 할 수 있습니다.
        만약 그 이전 버전이거나 다른 문제가 있을 경우 아래와 같이 합니다.
        #mount -t  iso9660 /dev/cdrom  /mnt/cdrom
```

——————————————————————————

2)실행중인 아파치 프로세스 찾기

/tmp 디렉토리로 APACHE , PHP , MYSQL을 다 옮겼다면 먼저 기존의 인스톨 때 생긴 APACHE를 지워야 합니다. 만약 있는 상태에서 또 인스톨 한다면 두개가 겹쳐 에러가 발생할 소지가 생깁니다. 리눅스 인스톨 때 APACHE를 제외했다면 이 부분은 넘어가도 됩니다.

#ps -ef | grep http - 이 명령을 실행하면 현재의 아파치 서버 프로세서를 확인할 수 있습니다.

3)실행 중인 아파치 프로세서 제거

Root 프로세서 541번을 제거합니다. 그러면 자식 프로세서도 다 같이 제거됩니다. 이 프로세서 번호는 다를 수 있습니다.

#kill 541 - 프로세서 제거 합니다.

실행할 때 마다 프로세서는 다르기 때문에 혹시 프로세서 번호가 자신의 것과 다르다고 이상히 여기지 마세요. 프로세서를 제거 했다면 2)번 과정을 한번 더 해서 완전히 제거 된 것을 확인 합니다.

4)기존에 설치된 아파치 찾기

rpm버전의 인스톨이 대부분이겠지만 rpm이 아닌 경우 아파치 관련 디렉토리를 찾아 제거해야 됩니다. rpm버전인 경우 일일이 관련 디렉토리를 찾아 제거해도 되지만 rpm 명령으로 간단히 제거 할 수 있습니다.

그림 31: 그림. 3-1 아파치 프로세서 찾기

그림 32: 그림. 3-2 아파치 프로세서 제거와 확인

그림 33: 그림. 3-3 rpm으로 설치된 아파치 찾기

#rpm -ql apache - rpm으로 설치된 아파치 디렉토리를 찾아 줍니다.

rpm 패키지로 설치된 아파치 디렉토리를 찾는 방법은 만약 rpm 삭제 명령으로 아파큥칠캬캬쳉≠┤ 酉适┘킵프淄各뺐嚥耳냐〉헨際뱄訝Ⓡ의 내용을 찾아 제거해야 되기 때문에 알아두기 바랍니다. rpm 패키지로 설치할 경우 아파치 서버의 모듈 추가와 기타 다른 설정에서 여러 문제점들이 나타날 수 있습니다. 비록 처음 접하고 다소 어렵게 느껴지더라도 바이너리 인스톨을 권장 합니다.

`<TIP>`——————————————————————————-

rpm으로 설치된 프로그램의 정보 보기 #rpm -qi 패키지명 - 해당 패키지의 존재 여부를 확인 합니다. 위 명령은 rpm으로 설치된 PHP나 Mysql을 찾을 경우도 마찬가지 입니다. 만약 의존성 문제로 인해 rpm 패키지 삭제명령(5)번 참고)이 실행 되지 않을 경우 위 명령으로 관련 디렉토리와 파일을 찾고 삭제 하면 됩니다. 여기서 설명할 TAR 패키지 설치의 경우 아파치 설치 디렉토리는 아래와 같습니다.

```
/usr/local/apache
/apache/bin
/apache/htdocs
/apache/cgi
/apache/conf
/apache/icons
/apache/logs
    ...
```

——————————————————————————

5)rpm으로 설치된 아파치 제거 하기

#rpm -e apache - rpm 설치 아파치를 제거해 줍니다.

5)번 명령을 한번 더 실행해서 패키지가 제거 되었는지 확인합니다. 만약 아파큥칠캬캬쳉【냐〉프淄刻年摸僞繞rror 메시지가 나올 것입니다.

3.2 A.P.M 설치하기

1) apache 파일 풀기

tmp 또는 여러분들이 정한 임의 디렉토리로 아파치를 copy 합니다. 다음 아래의 명령을 실행 합니다.

#zcat apache_1.3.12.tar.Z | tar xvf - - 해당 패키지의 압축을 풀어 줍니다.

`<TIP>`——————————————————————————-

tar.gz 과 tar.Z 패키지 풀기

```
위 패키지를 푸는 방법은 몇 가지가 있습니다.
    #tar -xvzf 패키지명   -   이 명령으로 두 종류의 패키지를 다 풀 수 있습니다.
    #gunzip 패키지명  -  gz 와 Z로 압축된 패키지를 풉니다.
    #gzip -d 패키지명 -  gz 와 Z로 압축된 패키지를 풉니다.
```

——————————————————————————

위를 보면 apache_1.3.12/ 디렉토리가 생성된걸 볼 수 있습니다. 설치하는 과정에서 mc(midnight command : 옛날 도스 시절 m과 유사한 유닉스 프로그램)를 사용해서 설치하는 분도 있을 것입니다. 디렉토리 사용에 copy나 기타 명령들이 다양하므로 리눅스의 텍스트 모드에 익숙치 않은 사람은 이 프로그램을 사용하기 바랍니다.

2) 아파치 디렉토리의 ./configure 와 ./Configure 실행

그림 34: 그림. 3-4 rpm으로 설치된 아파치 제거하기

그림 35: 그림.3-5 아파치 패키지 풀기

그림 36: 그림.3-6 아파치 패키지 풀림 확인

그림 37: 그림. 3-7 ./configure 실행 화면

apache_1.3.12 디렉토리로 가서 ./configure를 한번 실행해 줍니다. Apache에 디렉토리 (./configure 와 ./Configure 두 가지가 있으니 주의하기 바랍니다.) 다음 /apache_1.3.12/src 디렉토리로 옮겨 ./Configure를 실행합니다.

3)Mysql 풀기

./configure , ./Configure를 했다면 이제 apache는 잠시 두고 mysql을 풀도록 하겠습니다. /tmp나 여러분이 정한 임의 디렉토리로 패키지를 copy 합니다. 그리고 다음 명령을 실행 합니다.

#zcat mysql-3.23.12c-alpha-pc-linux-gnu-i686.tar.gz | tar xvf - - 해당 mysql 패키지를 풀어 줍니다.

여기 인스톨 할 mysql은 3.23.12c 버전 입니다. 여러분은 최근 버전을 다운 받아 설치하세요. 이렇게 바이너리 배포본을 인스톨 하는 이유는 다른 것보다 어느 정도 성능 향상이 있기 때문입니다. 바이너리를 컴파일 할 경우는 정적으로 링크가 이루어지게 되기 때문입니다. 직접 컴파일할 경우라도 정적으로 링크시킬 수 있지만 구지 그럴 필요는 없고 바이너리 배포본을 사용하기 바랍니다.

4) 풀린 Mysql 패키지 옮기기

압축을 다 풀었으면 mc나 cp 명령으로 mysql 디렉토리를 /usr/local/mysql로 옮깁니다. 물론 mysql 디렉토리는 새로 만드는 것입니다. 필자의 경우는 mc로 간단하게 옮겼습니다.

위 그림을 보면 /usr/local/mysql 디렉토리로 옮겨 놓은 것을 볼 수 있습니다.

5) ./scripts/mysql_install_db 실행

그림 38: 그림. 3-8 /src 디렉토리의 ./Configure 실행 화면

그림 39: 그림. 3-9 mysql 패키지 풀기

그림 40: 그림. 3-10 mysql 패키지 풀림 화면

그림 41: 그림. 3-11 mc로 Mysql 옮긴 화면

그림 42: 그림. 3-12 ./scripts/mysql_install_db 실행

이제 /usr/local/mysql 디렉토리 안에서 ./scripts/mysql_install_db를 실행해 줍니다.

(*주의 : /usr/local/mysql/scripts 디렉토리에서 실행하는 것이 아님을 명심하기 바랍니다.)

6)Mysql Test 및 설정

./scripts/mysql_install_db 명령을 실행한 후 제대로 설치되었다면 이제 mysql을 간단히 테스트 해봅니다. 먼저, mysql의 사용자 패스워드를 바꾸어 줍니다. 디폴트는 패스워드 없이 root 권한이기 때문에 차후 php 스크립트를 만들더라도 이 부분은 꼭 집고 넘어가야 됩니다. /usr/local/mysql/bin 디렉토리에서 다음 명령을 실행해서 패스워드를 바꿉니다.

<TIP>————————————————————-

Mysql 리눅스와 함께 실행 하기

```
만약 여러분들이 리눅스 서버의 실행과 동시에 Mysql을 가동 하기를 원한다면 아래와 같이 etc/rc.d/rc.local
```

```
    rc.local 추가 부분
      if [ -f /usr/local/mysql/bin/safe_mysqld ]; then
        /usr/local/mysql/bin/safe_mysqld &
      fi
```

————————————————————

그림 43: 그림. 3-13 ./scripts/mysql_install_db 실행 결과 화면

① 패스워드 바꾸기

```
#./mysqladmin -u root password ‘  새로운 password ’

패스워드를 바꾸기 전 확인해보시면 알겠지만 #./mysql -u root 하면 그냥 접속 됩니다. 우선 암호를 만드
```

② Mysql 가동과 정지 하기

```
mysql에 관한 보다 상세한 정보는 뒤 mysql 부분을 참고 하세요.

#./safe_mysqld &  -  usr/local/mysql/bin 디렉토리 안에 safe_mysqld 파일이 존재하고 mysql을 백그라운
#./mysqladmin shutdown  -  Mysql 서버를 정지 합니다.
#./mysql.server start  -  /usr/local/mysql/sbin 디렉의 mysql.server 스크립트를 이용해 mysql을 실행
#./mysql.server stop  -  Mysql 서버를 정지 합니다.
```

③ ./mysql.server의 접속 확인

```
해당 스크립트를 사용하기 편한 곳으로 copy한 다음 모드를 chmod 700으로 바꾸어 사용 할 수도 있습니다.
( 필자의 경우 /sbin 디렉토리에 옮겨서 실행 했습니다.)
```

그럼 이제 mysql 서버는 동작하는 상태이므로 직접 접속을 해보겠습니다. 먼저 앞에서 말했던 패스워드 설정 부분을 다시 확인 하겠습니다.

위와 같이 바꾸고 난후 접속 합니다.(여러분들은 자신의 패스워드를 입력 합니다.)

위에서 -p 옵션을 줌으로써 패스워드 입력란이 뜹니다. 그러면 직접 입력하고 엔터하면 접속이 이루어 집니다. 다음 mysql> 바뀌게 됩니다.

④ Mysql 접속 후 기본적인 확인

```
여기에서 mysql의 database와 table를 확인해 보겠습니다.
Mysql에 직접 접속해서 서버와 대화하는 방법은 뒷 부분의 php에서 지원하는 mysql 함수와 연관됩니다.
따라서 mysql 대화법을 안다면 php에서의 mysql 함수 또한 간단히 해결 됩니다.
Mysql은 php와의 조화가 apache와 만날 때 더 강력해 짐은 아무도 부정하지 않을 것입니다.
비록 php와 apache만을 이용해서 데이터베이스 없이도 사용 할 수 있지만 mysql 데이터베이스가 접목 된다
더 강력해지는 것입니다.  요즘의 경우 대부분의 서버가 데이터베이스를 이용하므로 아주 간단한 bbs에서
대형 bbs에 이르기 까지 유용하게 사용됩니다.
게다가 mysql은 공짜 데이터베이스이기 때문에 더욱 인기가 치솟고 있습니다.

mysql>show databases ;  -  현재 생성 되어 있는 데이터베이스를 보여 줍니다.
               (여기서 그냥 show tables ; 명령을 실행할 경우 에러 메시지가 출력 됩니다.
                특정 데이터베이스 사용을 명시하지 않았기 때문입니다.)

mysql>use mysql ;  -  mysql 데이터베이스 사용을 명시 합니다.
                특정 데이터베이스 사용을 위해 use 명령을 사용 합니다.

mysql>show tables ;  -  디폴트로 만들어진 테이블을 확인할 수 있습니다.
                여기에는 사용자와 함수들 호스트명 등 mysql 전반에 관한 사항이 존재합니다


mysql>/s  -  세부적인 mysql 정보를 확인 합니다.
```

그림 44: 그림. 3-14 mysql.server 모드 변경 화면

그림 45: 그림.3-15 mysql.server 실행 화면

그림 46: 그림. 3-16 mysql passwd 변경 화면

그림 47: 그림. 3-17 mysql 서버 접속 화면

그림 48: 그림. 3-18 기본 데이터베이스와 테이블 확인

그림 49: 그림. 3-19 mysql 세부 정보 화면

현재 설치된 mysql의 버전과 사용자 프로토콜 등의 정보를 확인 할 수 있습니다.

7) PHP 풀기

자 그럼 이제 php 모듈을 apache에 접목시키도록 하겠습니다. 여기 사용되어질 php는 PHP 4 zend beta 3 버전 입니다. 현재의 경우 PHP 4 정식 버전이 나왔고 그리고 zend 옵티마이져 등이 나온 상태 입니다. 여러분들은 이것을 설치하기 바랍니다. 아직 불안정한 상태라면 조금 더 지켜봐도 되지만 패치와 옵티마이져 등이 있으므로 더 강력해진 것은 사실 입니다. 먼저 php을 copy 한 /tmp(임의 디렉토리) 디렉토리에서 압축을 풉니다. (PHP 4.0 과 설치 방법은 같으므로 진행 순서를 지켜 인스톨 하면 됩니다.)

```
#zcat  |tar xvf - - 해당 " " 패키지의 압축을 푸는 명령어
```

8) PHP ./configure 실행

"7)" 과 같이 압축을 풀고 난 후 /tmp/php-4.0b3 디렉토리에서 mysql과 apache 항목을enable 시키기 위해서 ./configure를 아래염걋決피瑢瞞森絳求墨厤

```
#./configure –with-apache=/tmp/apache_1.3.12 –with-mysql=/usr/local/mysql
```

이 명령은 나머지 다른 항목을 추가 할 경우와 같이 이어서 나열하면 됩니다.

이 같은 방법은 ./setup 설정 파일이 php3 버전에서는 안정 버전이므로 존재했지만, 개발 버전이나 아직 진행중인 버전일 경우 직접 configure 해 주어야 합니다. ./setup 설정의 경우는 그냥 실행 하면 대화식으로

그림 50: 그림. 3-20 PHP 패키지 풀기 명령어

그림 51: 그림. gd-1 gd 패키지 풀림 화면

설정을 하는 것입니다. php 인스톨 시 추가할 항목들을 yes , no로 답하는 것입니다. 여기서의 설정은 configure로 직접 설정하도록 하겠습니다. 만약 위 configure가 실패했을경우는 (apache와 mysql 지원 설정) php-4.0b 디렉토리 안에 존재하는 config.chche 파일을 제거하고 다시 -with 옵션을 사용해서 configure 해야 합니다. 일단 configure 을 실행 했다면(apache와 mysql 지원설정) 다음의 항목을 확인하는 방법으로 실패 여부를 체크 할 수 있습니다.

<TIP>——————————————————————————-

GD 설치 하기

Gd는 php에서 그래픽 함수를 사용하기 위해 설치 하는 것입니다. 이것을 사용하기 위해서는 php의 ./configure 실행 때 추가 해야합니다.("8)" 항목)

① tmp 디렉토리에 gd 패키지를 copy 하고 압축을 풉니다.

② gd make 하기

압축을 푼 임의의 gd 디렉토리에서 컴파일 합니다.

그림 52: 그림. gd-2 gd 컴파일 화면

그림 53: 그림. gd-3 gd를 임의 디렉토리로 옮기기

③ gd 옮기기

컴파일한 gd를 임의의 디렉토리로 옮깁니다. (필자는 /usr/lib로 옮겼습니다.)

④ php의 ./configure 실행

php를 ./configure 할 때 TEL∞ --with-gd=/usr/lib" 항목을 추가합니다.

설치 성공 여부는 체크 항목을 찾아 확인합니다.

위 그림에서 아파치 지원 여부를 확인 할 수 있습니다.

위 그림에서 mysql 지원 여부를 확인 할 수 있습니다. 다른 항목을 추가 하려면 -with 옵션으로 다른 부분을 더 추가 할 수 있습니다. 여러분들 중 혹 나중에라도 다시 PHP에 또 다른 지원을 추가할 경우가 생길 경우 해당 패키지를 인스톨 하고 php의 ./configure 실행시에 –with 옵션을 이용해서 기존의 항목과 새로 추가할 항목을 다시 나열해서 해주면 되고 설치 8)번 항목 이후를 반복 하면 됩니다.

9) PHP 컴파일

php의 ./configure를 모두 마쳤습니다. 그럼 본격적으로 PHP를 컴파일 하는 것을 보겠습니다. 임시 디렉토리의 PHP를 /usr/local/php 디렉토리로 복사해서 configure했다면 거기서 make를 실행하면 됩니다.

그림 54: 그림. gd-4 php의 ./configure 실행 시 체크 화면

그림 55: 그림. 3- 21 php의 ./configure 실행 시 apache인식 화면

그림 56: 그림. 3-22 php의 ./configure 실행 시 mysql 인식 화면

그림 57: 그림. 3-23 php의 ./configure 실행을 한 후 마지막 화면

그림 58: 그림. 3-24 해당 php 디렉토리에서 make 실행

(혹, 필자처럼 /tmp/php-4.0b3 디렉토리에서 make를 해도 무방합니다.)

#make 실행

10) PHP 인스톨

먼저 PHP를 인스톨 하기 전에 apache 디렉토리의 ./configure를 실행해 주고 install을 해야 합니다.
즉, 이것은 php를 인스톨 할 때 apache 디렉토리에서 필요한 파일을 참조 시키기 위함입니다.
/php-4.0b3 디렉토리에서 아래와 같이 명령합니다.

```
    #make install   -  PHP를 인스톨 합니다.
```

그러면 아파치의 src/modules 디렉토리로 모듈이 만들어져 이동합니다.

11) 모듈 확인과 아파치 Configuration 파일을 수정

```
    #ls -l   -   아파치 디렉토리밑에 /src/modules/php4  디렉토리의 내용을 살펴 봅니다.
```

위 그림에서 아파치 디렉토리에 php4 디렉토리가 생성되고 libphp4.module 파일과 나머지 파일들이 생성되는 것을 볼 수 있습니다. (PHP 4.0 의 경우도 마찬가지 입니다.) 다음 tmp/apache_1.3.12/src 디렉토

그림 59: 그림. 3-25 make 실행 후 화면

그림 60: 그림. 3-26 아파치 디렉토리 /src/modules/php4의 내용

그림 61: 그림. 3-27 Configuration파일의 수정 화면

리로 이동합니다. 그리고 Configuration 파일의 내용에 AddModule modules/php4/libphp4.a를 추가해 줍니다.

(*주의 : configure 파일이 아닌 Configuration 파일을 수정해야 됩니다. 또한 libphp4.a는 존재하지 않는 파일이므로 이대로 적어 주세요.)

12) 아파치 Configure 실행과 make 하기

Configuration 파일의 수정을 끝내고 수정 항목의 인식을 위해 /tmp/apache_1.3.12/src/ 디렉토리에서 ./Configure를 다시 실행합니다.

다음 /tmp/apache_1.3.12/src 디렉토리에서 make를 실행합니다.

위에서 중간 부분에 .. o php4_module uses ConfigStart/End 를 확인할 수 있습니다.

위 사항이 없을 경우 다시 뒤의 항목들을 확인하기 바랍니다. (Configuration 추가 부분의 문제일 가능성이 큽니다.)

대부분의 경우는 맞게 진행 되지만 간혹 실수를 할 수도 있기 때문에 주의하기 바랍니다. Apache , php , mysql의 설치는 그다지 어려운 부분이 아니므로 모든 것이 디폴트인 상황에서 대부분 성공적으로 인스톨 될 것입니다.

13) 아파치 install

그림 62: 그림. 3-28 ./Configure 실행과 make 실행

그림 63: 그림. 3-29 make 실행 화면과 make install 명령 화면

make 과정을 마쳤다면 /tmp/apache_1.3.12 디렉토리로 이동 합니다.

그리고 make install을 실행합니다.

#make install

"13)" 을 마지막으로 모든 설치는 끝났지만 아직 php를 사용하기 위해서는 몇 가지 설정할 부분이 더 남아 있습니다.

Php make install의 마지막 문구를 보면 /usr/local/apache/conf/httpd.conf의 서버 설정을 변경해서 PHP 스크립트를 사용해야 함을 알 수 있습니다.

14)아파치의 httpd.conf 파일 수정

먼저 /usr/local/apache/conf 디렉토리로 이동 합니다.

그리고 편집기를 이용해서 httpd.conf 파일을 수정합니다. 수정 하는 부분은 아파치에서 php 확장자를 인식시키기 위함입니다. php 말고도 여러 가지 확장자를 설정할 수도 있습니다.

AddType application/x-httpd-php .php .html .htm .inc

이렇게 다른 확장자를 추가합니다. 즉, php 언어를 html이나 기타 다른 확장자 파일로도 사용 가능하게 해줍니다.

이렇게 모든 설정을 마쳤습니다. 이제 확인할 부분은 apache의 정상적인 작동 유무와 모듈들의 정상적인

그림 64: 그림. 3-30 PHP make install 결과 화면

그림 65: 그림. 3-31 아파치의 httpd.conf 파일 수정

동작 유무입니다.

Apache_1.3.12 같은 경우 httpd.conf의 설정이 전 버전 보다 조금 다를 수도 있지만 기본적인 설정은 같습니다.

(http.conf 설정에서 php4설정 부분이 추가되어 있습니다.)

15) 아파치 작동 유무 확인

아파치를 실행하는 방법과 멈추는 방법 등의 여러 가지 옵션을 확인해 보겠습니다.

apachectl 사용하기

apachectl 는 /usr/local/apache/bin 디렉토리에 서버와 관련된 유틸리티들이 존재합니다.

./apachectl (start|stop|restart|fullstatus|status|gracefull|configtest|help)

```
     -. start - httpd 데몬을 띄웁니다.
     -. stop - httpd 데몬을 종료 합니다.
     -. restart - httpd 데몬이 존재 하면 다시 시작하고 없을 경우 새로 시작 합니다.
     -. fullstatus - mod_status가 사용 가능할 경우 서버의 상태를 자세히 보여 줍니다.
     -. status - mod_status가 사용 가능할 경우 서버의 상태를 간략히 보여 줍니다.
     -. configtest - httpd.conf 파일을 변경 했을 경우 이 내용을 테스트 합니다.
     -. help - 도움말을 보여 줍니다.
```

#./apachectl start - httpd 데몬을 실행 합니다.

#./apachectl stop - httpd 데몬을 종료 합니다.

<TIP>————————————————————————-

pache를 리눅스와 같이 실행하기

```
Mysql과 같이 시작 설정에 추가해 주면 됩니다.
"/etc/rc.d/rc.local" 파일에 아래 항목을 추가 합니다.

    If [ -f /usr/local/apache/bin/httpd ]; then
          /usr/local/apache/bin/apachectl start
       fi
```

————————————————————————————————-

이제 httpd 데몬을 띄우고 정상적으로 동작하는지 확인합니다. 확인하는 방법은 브라우저를 통해서 해도 되고, 텍스트 모드에서 #ps -aux | grep apache를 입력해서 확인하는 방법이 있습니다. 먼저, 브라우저 같은 경우, 확인 하는 곳이 서버면 http://localhost를 입력 하면 되고 DNS 설정이 되어 있고 등록된 상태라면 해당 DNS를 입력 하면 됩니다. 만약 확인하려는 곳이 서버가 아니고 클라이언트 PC일 경우 아파치 서버의 IP나 해당 DNS를 입력합니다.

위의 경우 클라이언트에서 아파치 서버로 접속한 예입니다. 아파치_1.3.12 버전의 경우 처음 브라우저로 접속하면 위와 같이 나오니 당황하지 마세요. 위의 화면은 /usr/local/apache/htdocs 디렉토리에 존재하는 index.html을 참조하는 것인데 httpd.conf 파일의 설정 중에 서버의 처음 보여줄 파일설정을 새롭게 해주면 그 화면만 출력되어 집니다. 아파치_1.3.9의 경우 index.html이 하나만 존재 하므로 그 파일만 출력되는 것입니다.(1.3.12의 경우 각 언어별 index 파일이 존재 합니다.)

위 그림은 여러 가지 index.html중에 English를 선택한 화면입니다.

16) PHP4 동작 확인과 각 모듈 인식 확인

이제 php4 와 mysql의 모듈들이 제대로 동작 하는지 확인해 보겠습니다. 먼저, mysql 서버를 실행했는지 확인하고, test.php를 만들어 웹 상으로 모듈들의 인식을 확인 하겠습니다. (phpinfo( ) 함수를 사용합니다. 이 함수는 PHP의 정보를 출력합니다.)

그림 66: 그림. 3-32 클라이언트 PC에서 초기 아파치 접속 화면

그림 67: 그림. 3-33 index.html.en 선택 화면

그림 68: 그림. 3-34 phpinfo() 함수 실행 화면

test.php 파일의 내용

```
<?php
    phpinfo();
?>
```

위의 파일을 편집기를 이용해 만들어 줍니다. 그리고 /usr/local/apache/htdocs 디렉토리에 copy 합니다.

(필자의 경우는 test 디렉토리를 따로 만들었습니다.)

그리고 브라우저를 실행해 봅니다. 위에서 말했듯이 브라우저를 실행 했을 경우 서버는 htdoc 디렉토리를 참조하기 때문에 리눅스 머신에서서는 http://localhost/test.php를 입력하면 됩니다.

(필자의 경우 http://192.168.0.1/test/test.php)

위 화면이 보이면 제대로 동작하는 것입니다. 그러나 다시 스크롤바를 내려 php 설정과 mysql 설정들을 확인하기 바랍니다. Php 설정의 경우 php.ini 파일을 이용하는데 위 화면에서 path:/usr/local/lib에 존재함을 알 수 있습니다.

<TIP>————————————————————————-

HP.INI 가 /usr/local/lib에 없을 경우

그림 69: 그림. 3-35 PHP.INI 파일

```
    php를 처음 풀고 컴파일 했던 디렉토리로 이동해 php.ini-dist 파일을 /usr/local/lib 디렉토리로 copy 하
```

————————————————————————————-

이렇게 모든 설정과 확인을 마쳤습니다. ("1)"   "16)" 번 항목은 몇번 되풀이 하면서 익히세요.)

PHP.INI 설정

PHP를 웹 서버에서 여러 방법으로 실행할 수 있게 설정하는 부분 입니다.

(php4 의 경우 php.ini 이고, php3의 경우 php3.ini 입니다.)

설정들을 수정 할 시에는 on, off 나 해당 값을 추가 또는 삭제하고 웹상에서 test 파일을 다시 확인하면 주어진 설정들이 바뀌었는지 확인할 수 있습니다.   다른 방법도 있겠지만 이 방법이 제일 확실합니다.

<TIP>———————————————————————-

리눅스에서 Zend 설치하기

Zend optimizer의 설치는 보기 보다 간단 합니다. 리눅스의 모든 패키지 파일은 README라는 파일이 존재 하면 이와 연관된 파일을 자세히 읽었던 사람 이라면 금방 설치 방법을 알아냈을 것입니다.

Zend optimizer는 PHP 4 beta 4 이후 버전만 사용 가능하며 지금 여러분들은 이미 PHP 4.0을 설치 했을 것입니다.

그림 70: 그림. 3-36 클라이언트에서 본 PHP.INI의 각 옵션 부분

지원하는 시스템

PHP 4 beta 4 버전의 경우

- Linux (glibc 2.1 systems including Redhat 6.1, Slackware 7.0, Suse 2.2, Mandrake 7.0)
- FreeBSD 3.4 (not tested on other versions of FreeBSD)
- Windows NT 4.0 (not tested on other versions of Windows)

설치와 인스톨

ZendOptimizer-Beta1-Linux-glibc2.1.tar.gz 파일을 풀고 /usr/local/Zend/lib 디렉토리를 생성 한다. 그리고 ZendOptimizer.so 파일을 이 디렉토리로 copy 한다.

다음 *PHP.INI* 파일에 아래의 내용을 추가 한다.

```
zend_optimizer.optimization_level=1
```

```
zend_extension="/usr/local/Zend/lib/ZendOptimizer.so"
```

위 level은 7번 까지 존재 하며 패키지 README를 읽어 보기 바랍니다.

동작 확인은 phpinfo() 함수를 사용해 PHP.INI 설정을 확인 해서 아래의 내용이 나오면 동작 하는 것입니다.

```
This program makes use of the Zend scripting language engine:
Zend Engine v0.90, Copyright (c) 1998, 1999 Andi Gutmans, Zeev Suraski
with Optimizer v0.95, Copyright (c) 1998-2000, by Zend Technologies Ltd.
```

PHP.INI 내용과 각 옵션의 기능(PHP4 기준)

설정 파일은 총 9개의 항목으로 나누어져 있으며 여기서는 함께 설명하겠습니다.

1. engine - PHP가 아파치 모듈로 설치 되었을 경우 특정 디렉토리에서만 스크립트가 동작하게 합니다. 이것은 아파치 httpd.conf 파일에 아래 항목을 추가합니다.

```
<Directory "해당 디렉토리">
php_admin_flag engine on(PHP3의 경우 php3_engine on)
</ Directory>
```

2. short_open_tag - <? ~ ?> php tag 사용여부를 정합니다. XML과 같이 사용할 경우 이 부분은 on으로 해야 합니다.

3. asp_tags - <% ~ %> ASP 태그의 사용 여부를 설정합니다.

4. y2k_compliance - y2k 문제에 해당하는 브라우저를 위한 설정입니다.

5. safe_mode - php를 안전 모드로 동작 하게끔 하는 설정입니다.

   safe_mode_exec_dir 지시자의 해당 디렉토리에 존재하는 실행 파일이 아닌 경우 실행 되지 않습니다. (root에 있는 파일만 참조하게 합니다.)

6. safe_mode_exec_dir - 실행 프로그램이 있는 디렉토리 명을 적어 줍니다.

"5번"참고

7. max_execution_time - 스크립트 실행 시간을 초단위로 설정합니다.

8. memory_limit - 스크립트 실행 때 메모리 사용을 설정합니다.

9. error_reporting - 에러 출력을 설정합니다.

에러의 종류

1 - 일반적인 에러 출력(Normal function errors)
2 - 일반적인 에러 출력(Normal warnings)
4 - 문법적인 에러 출력(Parser errors)
8 - 버그의 가능성이 있으며 무시할 수 있는 에러 출력(Notices)

10. display_errors - 에러 상황을 html 양식으로 보여줍니다.

11. log_errors - 여러 가지 에러를 로그 파일에 저장합니다.

12. track_errors - $php_errormsg에 마지막의 에러나 경고 메시지를 저장합니다.

13. warn_plus_overloading - 스크립트에서 문자열 결합사용 때 " . " 연산자를 사용하지만 만약 " + "를 사용했을 경우 경고 메시지 출력 여부를 설정합니다.

14. track_vars - GET, POST, COOKIE 등으로 전달된 변수 값을 각각의 $HTTP_GET_VARS , $HTTP_POST_VARS , $HTTP_COOKIE_VARS의 배열 변수로 자동 저장 여부를 설정합니다.

15. magic_quotes_gpc - GET , POST , COOKIE 등으로 전달된 데이터에 ‘ " / 등의 문자가 존재할 경우 자동으로 (\)역슬래쉬를 붙일지 여부를 설정합니다. 이것은 각 문자를 스크립트와 구별해서 인식시키기 위해 필요한 부분입니다.

16. magic_quotes_runtime - 외부 스크립트나 여타 소스로부터 돌아오는 데이터에 대해 (\)역슬래쉬를 붙일지 여부를 설정합니다.

17. auto_prepend_file - INCLUDE와 비슷한 부분이고, 스크립트의 처음 부분에 포함될 파일명을 적어 줍니다.

18. auto_append_file - "17번"과 같으며 마지막 부분에 포함될 파일명을 적어 줍니다.

19. include_path - INCLUDE, REQUIRE 문의 파일 경로를 설정합니다.

20. doc_root - 서버의 루트 디렉토리를 설정합니다. 안전 모드로 동작할 경우 해당 디렉토리를 벗어나 다른 쪽으로 접근을 할 수 없습니다.

21. user_dir - 서버의 계정을 가진 일반 유저에 PHP 스크립트의 사용을 허가할 수 있습니다.

22. upload_tmp_dir - 파일이 업로드 될 디렉토리를 설정합니다. 기본적으로 /tmp 디렉토리로 업로드 됩니다.

23. upload_max_filesize - 업로드 할 수 있는 최대 파일 크기를 설정합니다.

24. SMTP - mail( ) 함수의 사용을 위해 현 서버의 DNS , IP를 적어줍니다.

25. sendmail_from - 보내는 메일 주소를 설정할 수 있습니다.

26. sendmail_path - sendmail 서버가 있는 위치를 적어 줍니다.

27. debugger.host - 디버거로 사용하는 호스트 명을 적어 줍니다.

28. debugger.port - 디버거로 사용하는 포트 번호를 적어 줍니다.

29. debugger.enabled - 디버거의 사용 여부를 설정 합니다.

30. session.save_handler - 세션과 해당 데이터를 저장하는 방식을 설정하는 부분입니다. 기본적으로 파일 방
입니다. 세션 항목은 이 책에 사용되는 회원제 부분에서 사용됩니다.

31. session.save_path - 세션 전달 인수를 정의 하는 부분이며 디폴트인 파일 방식일 경우 해당 파일의 경로를
적어 줍니다.

32. session.name - 쿠키 이름으로 사용되어질 세션 이름을 설정합니다. 디폴트는 PHPSESSID 입니다.

33. session.auto_start - 세션 요청 시에 자동 모듈 생성 여부를 설정합니다.

0 - off , 1 - on

34. session.lifetime - 해당 세션이 유지 되는 시간을 설정 합니다.
디폴트의 경우 0이며 사용자의 브라우저가 종료 될 때까지 유지됩니다.

35. session.gc_probability - 해당 세션 값이 제거 될 경우 낭비 되는 시스템 메모리 자원을 반납하는 시점을
설정합니다.

36. session.gc_maxlifetime - 해당 세션의 시스템 메모리 자원 반납을 가질 수 있는 시간을 설정합니다.

다음 장에서는 리눅스에서 A.P.M의 설치 과정을 보았습니다. 다음 장에서는 윈도우 환경에서 Apache, Php, Mysql의 설치를 설명하겠습니다.

제 4 절 윈도우 98과 2000(NT)에서 A.P.M 설치

윈도우 98과 윈도우 2000 그리고 NT에서 APACHE , PHP, MYSQL을 설치하는 방법을 살펴 보도록 하겠습니다. Linux와 sloraris에서의

설치 보다는 한결 간편한 부분들이 많고 손쉽게 클릭과 몇 가지 수정 만으로도 간단히 설긍칠캭캭쳉÷肩獰沮埧求墨麻

하지만 아직 최신 버전의 apache의 경우나 mysql , php , zend Optimizer 경우는 검증이 되지 않은 상태이므로 윈도우 환경에서의 서버 운영은 아직 무리라고 봅니다. 윈도우 환경에서 간편히 작업하고 그리고 스크립트 테스트하는 것에는 무리가 없으므로 그렇게 활용해도 됩니다.

4.1 윈도우용 A.P.M 설치

1) mysql 설치(3.22.34win)

그림 71: 그림. 4-1 Mysql 인스톨 화면

원하는 곳에 mysql을 풀고 인스톨 합니다.

대부분의 일반 인스톨 방법처럼 3가지 인스톨 선택 방법이 있으나 전형적인 방법으로 설치하면 됩니다.

c:\mysql 디렉토리로 이동한 후 제대로 인스톨이 되었나 확인해 봅니다.

2) ODBC 인스톨

ODBC의 경우 nt(2000)용과 98용이 따로 존재하므로 해당 ODBC를 인스톨하면 됩니다.

3) PHP 인스톨 (PHP4 RC1)

PHP의 경우 인스톨러가 없기 때문에 특정 디렉토리를 생성하고 거기에 압축을 풀어 놓아야 됩니다. 여기서는 c:\php 디렉토리에 압축을 풀었습니다.

사용자에 따라 특정 디렉토리로 하면 됩니다.

위 그림에서 c:\php 디렉토리를 보면 php.ini-dist 파일이 존재 하는데 이 파일을 php.ini로 고친 후 (윈도우 98의 경우는 c:\windows에 NT나 2000의 경우는 c:\winnt로 copy 해야 됩니다.) 다음 환경 설정 파일(php.ini)을 메모장 등으로 내용을 수정합니다.

extension_dir = "c:\php"로 수정 합니다.

다음 그림 4-7에서 extension 항목도 수정합니다 enable는 앞의 ;를 제거합니다.

4) APACHE 인스톨 (1.3.12win)

그림 72: 그림. 4-2 인스톨 확인

그림 73: 그림. 4-3 ODBC 인스톨 화면

그림 74: 그림. 4-4 PHP 인스톨 디렉토리 확인

그림 75: 그림. 4-5 PHP.INI 위치 화면

그림 76: 그림. 4-6 PHP.INI 파일

그림 77: 그림. 4-7 PHP.INI extension 항목

그림 78: 그림. 4-8 아파치 인스톨 화면

아파치 역시 전형적인 인스톨을 하면 됩니다. 인스톨을 다 하면 c:\program Files\apache Group\apache로 디렉토리가 생성 됩니다.

Linux 에서와 마찬가지로 conf 디렉토리의 httpd.conf 파일에 addtype 항목을 추가 합니다.

원하는 확장자를 아래와 같이 추가해주면 됩니다.

```
ScriptAlias /php/ "c:/php/"

AddType application/x-httpd-php .inc
AddType application/x-httpd-php .php3
AddType application/x-httpd-php .php
AddType application/x-httpd-php .htm
AddType application/x-httpd-php .html
AddType application/x-httpd-php .phtml
AddType application/x-httpd-php-source .phps

Action application/x-httpd-php "/php/php.exe"
```

위에서 addtype 항목은 여러분들이 필요한 만큼 확장자를 추가해 줍니다.

그림 79: 그림. 4-9 아파치 생성 디렉토리

그림 80: 그림. 4-10 아파치 httpd.conf 파일

그림 81: 그림. 4-11 Zend 복사할 위치

5) ZEND Optimizer 설치 (beta2)

가속기의 설치는 먼저 해당 파일을 c:\Program File\Zend\lib 디렉토리에 복사합니다.

다음 아래 항목을 php.ini 파일에 추가합니다.

```
zend_optimizer.optimization_level=7
zend_extension_ts="C:\Program Files\Zend\lib\ZendOptimizer.dll"
```

이제 모든 설치가 완료 되었습니다.

여기서 다시 c:\php 디렉토리를 점검해 봅니다. 최신 버전일수록 앞에서 설정한 부분과 다를 수 있습니다. 위의 절대 디렉토리에 php에 해당하는 모든 파일을 저장해야 됩니다. 물론 앞에서의 설정이 c:\php이기 때문입니다.

이 모든 설치의 정상 동작 확인은 간단한 phpinfo() 함수를 이용해서 스크립트로 확인해 보는 겁니다. 아래의 파일은 리눅스에서 test.php와 같습니다.

```
<?php
    phpinfo();
?>
```

그림 82: 그림. 4-12 PHP.INI 파일에 Zend 부분 추가

참고. 윈도우용 mysql을 설치 한 후 데몬을 띄우는 방법에서 많은 부분 어려움을 토하는 사람들이 있을 줄 압니다. 저도 그랬으니까요… 자세한 방법은 mysql 사이트의 도큐먼트 란에 가면 운영체제 별로 보다 자세한 방법이 있으며 참고로 윈도우 98의 경우 bin디렉토리 밑에 존재하는 mysqld-XXXX 파일의 이름을 mysqld 수정 후 standalone 옵션으로 실행 하면 될것이다.. 의문이 있으면 직접 사이트를 방문해서 더 자세한 내용을 찾아 보기 바랍니다. http://www.mysql.com/mysql/full/

다음 장에서는 Solaris에서의 Apache , Php , Mysql의 설치를 설명하겠습니다.

이장은 개인적으로 제외 했습니다…. 의문점이 있으면 메일로 질문하기 바랍니다.

제 5 절 솔라리스에서 APM설치하기(제외)

제 6 절 APACHE 설정과 고급 기능

이 장에서는 아파치의 여러 가지 설정과 고급 기능에 대해 설명하겠습니다. httpd.conf 파일이 설정에 관련된 모든 정보를 가지고 있으므로 이 파일의 옵션을 위주로 설명하겠습니다.

6.1 아파치의 구성

아파치를 성공적으로 설치했다면 /usr/local/apache/에 여러 가지 디렉토리가 생성됩니다. 사용자에 따라 이 디렉토리는 다를 수 있습니다. 우선 기본적인 디렉토리만 나열했습니다.

1) 디렉토리 구성

```
/bin 디렉토리 - apache 사용 시에 필요한 유틸리티들이 들어 있습니다.
/cgi - bin 디렉토리 - cgi 스크립트가 있는 곳입니다.
/conf 디렉토리 - apache 서버의 여러 가지 설정 파일들이 있습니다.(httpd.conf)
/icons 디렉토리 - apache 서버에 사용되는 아이콘들이 들어 있습니다
/logs 디렉토리 - apache 서버 사용 시 발생하는 여러가지 log들이 들어 있습니다.
```

```
/conf 안에 httpd.conf 파일의 경우 소스에서 다시 copy 해서 사용 가능하므로 나중이라도 설정 잘못으로 apach
다시 인스톨 하는 일이 없도록 하세요.
```

2) httpd.conf 구성

그러면 httpd.conf 파일의 내용을 한번 살펴보면서 설정을 설명하겠습니다. 먼저 이 파일은 apache 전반에 관한 설정이 들어 있고 간단히 옵션만 고쳐서 바꿀 수 있습니다. 고치고 나면 당연히 restart 해야 합니다. 서두 부분은 이 파일의 내용을 담고 있습니다. 먼저 이 파일은 총 3가지로 구분할 수 있습니다.

```
1. 아파치를 전반적으로 제어하는 부분
2. 가상호스트에서 처리되지 않는 요청을 모두 처리하는 주 서버 또는 기본 서버의 작동을 제어하는 부분
3. 다른 IP와 호스트 이름을 처리하는 가상호스트 설정 부분
```

apache 1.3.4 이전 버전의 경우 3가지 파일(httpd.conf, srm.conf, access.conf)등의 파일이 존재했지만 지금은 httpd.conf 파일로 통합되어진 상태입니다.

6.2 httpd.conf의 관리

6.2.1 아파치 서버의 전반적인 제어(Section 1)

#서버 타입 - 서버의 시작을 어떻게 할 것인지 설정하는 부분입니다. 두 가지가 있는 데 대부분 Standalone 방식을 사용합니다. Inetd 방식은 새로운 프로세서를 만들 때마다 설정 파일을 참고 하는 단점이 있고 웹

그림 83: 그림. 6-1 서버 타입 설정

그림 84: 그림. 6-2 서버 루트 설정

서버를 restart 안 해도 됩니다.

Standalone 방식의 경우 restart 해줘야 하지만 부하에 보다 안정적 입니다.

#서버 루트 - 이 설정은 서버의 설정, 에러, 로그 파일이 기록되는 디렉토리를 말합니다. 위 그림의 /usr/local/apache 가 디폴트 경로 입니다.

#Lockfile - 이 설정은 아파치 컴파일을 USE_FCNTL_SERIALIZED_ACCEPT 나 USE_FLOCK_SERIALZED_ACCEPT 의 옵션으로 했을 경우 잠금 파일의 경로를 설정할 때 사용합니다. 디폴트는 사용 안함으로 설정되어 있습니다. 로그 디렉토리가 NFS 마운트 된 경로로 되어야 이 값을 바꾸어 줍니다.

#PidFile - 웹 서버가 시작할 때 Process id num를 기록하는 파일입니다.

#ScoreBoardFile - 내부 서버 프로세서의 정보를 저장하는 파일 입니다. 그 경로 명과 같이 적혀져 있습니다. 특정 아키텍처에서 필요할 경우 서로 다른 아파치 웹 서버 데몬이 같은 파일을 공유하지 않아야 합니다.

#ResourceCofig / AccessConfig - apache 1.3.4 이전의 버전에서는 srm , access conf

파일을 httpd.conf과 함께 읽었지만 이후엔 httpd.conf 파일에 한데 뭉쳐져 관리하기 편해 졌습니다.

서버가 이 파일의 내용을 무시하게 하기 위해서는 "dev/null"(Unix) 또는 "nul"(Win32) 값을 지정 해야 합니다.

그림 85: 그림.6-3 Lockfile 설정

그림 86: 그림.6-4 Pidfile 설정

그림 87: 그림.6-5 ScoreBoardFile 설정

그림 88: 그림.6-6 ResourceCofig / AccessConfig 설정

그림 89: 그림.6-7 Timeout 설정

그림 90: 그림. 6-8 KeepAlive 설정

#Timeout - 클라이언트가 서버에 요청한 정보를 받을 때 소요 되는 시간을 정해 줍니다.

디폴트는 300으로 설정되어져 있고 초단위로 설정을 합니다. 서버 성능에 따라 적당한 값을 적어 주면 효율적 입니다.

#KeepAlive - 디폴트 값이 on으로 되어 있습니다. 그 이유는 성능저하 때문인데 자식 프로세서가 한가지 일만 처리하고 죽는 것이 아니라 다음의 일도 처리함으로써 성능 향상을 꾀하는 것입니다.

#MaxKeepAliveTimeout - KeepAlive가 설정되어 있을 때, 클라이언트의 요청을 몇 번 이나 처리하고 다시 처음 상태로 돌아 가는지를 설정 합니다. 만약 이 설정이 "0"일 경우 클라이언트가 접속을 끊을 때까지 계속 연결 상태로 있습니다.

(대부분의 경우 높은 성능 향상을 위해 높은 값을 적어 줍니다.)

#KeepAliveTimeout - KeepAlive 가 설정되어 있을 경우 클라이언트의 요청이 정해진 초안에 없을 경우 접속을 끊습니다.

여러분들의 서버에 맞게 초를 바꾸어 주면 됩니다.

#MinSpareServer , MaxSpareServer - 서버 풀을 설정하는 부분입니다. 여기 설정된 디폴트 값은 대부분의 서버에 적합한 상태입니다.

즉, 클라이언트의 요청이 많은 사이트라면 적당히 이 값들을 증가 또는 감소해서 사용해야 합니다. 아파치는 최소 값과 최대 값을 정해 줌으로써 클라이언트의 요청이 많은 경우든 작은 경우든 쉽게 프로세서를

그림 91: 그림. 6-9 MaxKeepAliveTimeout 설정

그림 92: 그림. 6-10 KeepAliveTimeout 설정

그림 93: 그림. 9-11 MinSpareServer , MaxSpareServer 설정

그림 94: 그림. 6-12 StartServer 설정

늘리고 줄일 수 있도록 하는 것입니다. 즉, 아파치는 클라이언트의 요청이 적을 경우 또는 최소 값 보다 프로세서가 적을 경우에 자동으로 최소 여유 서버의 개수를 맞추도록 노력 합니다. 최대 값보다 프로세서가 많을 경우라면 자동적으로 최대 여유 서버의 개수에 맞추어 줍니다.

#StartServer - 아피치 서버를 실행 할 때 같이 생성될 여분의 프로세서의 개수를 정합니다.

디폴트 값은 5 입니다.(6개 생성되어짐)

#MaxClients - 동시에 아파치 서버에 접속할 수 있는 클라이언트의 개수를 설정 합니다.

이 값을 늘리거나 줄일 경우 MinSpareServer , MaxSpareServer ,StartServer 의 값도 조정 해 주어야 합니다. 즉, 최대값을 넘어설 嚥里Φ瓚潔青®의 요청은 앞의 요청이 끝날 때까지 대기 상태로 있어야 합니다. 이 설정은 아파치 서버가 많은 자원을 낭비해 서버 부하가 생기는 것을 막기 위해 사용됩니다.

#MaxRequestsChild - 각각의 자식 프로세서가 죽기 전까지 처리할 수 있는 요청 수를 설정 합니다. 대부분의 시스템에서는 하나의 프로세서가 오랫동안 사용되면 메모리 또는 자원 누출 현상이 발생 할 수 있으므로 이 프로세서는 자동으로 제거 됩니다. 하지만 솔라리스 운영체제의 경우 이 설정이 필요합니다.

#Listen - 아피치를 특정IP 또는 포트와 결합 시킬 경우 사용 합니다.

<VirtualHost> 지시자 또한 참고 하세요.

#BindAddress - 이 옵션을 사용해서 가상호스트를 지원할 수 있습니다. 옵션에는 '*' 'IP address' '인터넷 도메인 이름'을 사용할 수 있습니다.

그림 95: 그림. 6-13 MaxClients 설정

그림 96: 그림. 6-14 MaxRequestsChild 설정

그림 97: 그림. 6-15 Listen 설정

그림 98: 그림. 6-16 BindAddress 설정

그림 99: 그림. 6-17 Dynamic Shared Object (DSO) 설정

<VirtualHost>과 Listen 지시자도 참고 하세요.

가상호스트에 관한 자세한 사항은 *http://kldp.org* `<http://kldp.org>`로 가시면 한글로 작성된 많은 문서들을 참고하실 수 있습니다.

Dynamic Shared Object (DSO) - DSO(동적 공유 객체)를 사용 하기 위해서 모듈을 지시하는 설정입니다. DSO의 작동 방식을 보려면 아파치 1.3 배포판의 README.DSO 파일을 참고 하기 바랍니다. 여러분들의 httpd 바이너리에 내장된 모듈 목록을 확인하려면 ℡¬◀◀∞http -l" 을 실행하면 볼 수 있습니다.

모듈 적재의 순서는 중요 사항이므로 아무렇게나 순서를 바꾸면 안됩니다.

#ExtendedStatus - "server-status"가 호출 되었을 때 아파큥칠캭캭쳉÷靡션羈버 상태 정보를 만들 것인지 여부를 설정합니다.

Off일 경우 기본적인 사항의 정보만 생성합니다.

6.2.2 아파치의 주 서버 설정(Section 2)

가상호스트에서 처리되지 않는 요청을 모두 처리하는 주 서버 또는 기본 서버의 작동을 제어하는 부분입니다.

#Port - 독립형 서버가 대기하는 포트를 설정합니다. 포트 번호가 1023번 보다 낮을 경우 처음 root 권한으

그림 100: 그림. 6-18 ExtendedStatus 설정

그림 101: 그림. 6-19 Port 설정

그림 102: 그림. 6-20 User , Group 설정

로 httpd가 실행되어야 합니다.

기본적으로 설정된 포트 외에 다른 포트를 설정함으로써 여러분들도 httpd를 운영할 수 있습니다.

<TIP>——————————————————————-

유동 ip로 공짜 도메인 사용하기

여러분들 대부분의 경우가 케이블 모뎀이나 adsl을 사용한다고 가정 하겠습니다. 일단 바뀌는 유동 ip에 공짜로 도메인을 주는 곳을 보면 *http://www.dyndns.org/* `<http://www.dyndns.org/>`- 여기의 경우 일단 가입을 한 후 24시간 안에 해당 id의 패스워드를 메일로 받습니다. 그리고 그 패스워드로 접속한 다음 원하는 패스워드로 바꾸어 줘야 합니다. 고정 ip의 경우도 도메인을 제공하고 여러 가지 옵션들이 다양하게 있습니다. 클라이언트를 받아 서버의 가동과 동시에 바뀌는 ip에 따른 도메인 설정을 잡아 줄 수도 있습니다. *http://xdomain.neti.org/* `<http://xdomain.neti.org/>`- 여기는 한국 사이트이고 접속이 자주 끊기는 현상이 있습니다. 하지만 공짜니 불만을 가질 수는 없겠죠? 일단 가입 즉시 사용 가능하고, 친절히 관리자가 게시판에 답도 해줍니다.

——————————————————————————-

#User , Group - 독립형 서버에서 사용되며, 일반적으로 서버에서 작업 프로세서는 nobody로 설정을 해 줍니다.

즉, 제일 권한이 없는 것으로 설정을 합니다. 보안 때문에 이렇게 설정을 하는 것이고, 만약 nobody로

그림 103: 그림. 6-21 ServerAdmin 설정

설정 되어진 상태라면 telnet으로의 접근 또한 막아야 됩니다. SCO(ODT 3)의 경우 User nouser , Group nogroup로 설정해 줍니다.

#ServerAdmin - 서버에 문제가 생겼을 경우 메일을 보낼 주소를 적어 주는 곳입니다.

이 주소는 에러 문서와 같이 서버가 생성하는 페이지에 나타납니다.

#ServerName - 만약 여러분들이 등록된 DNS를 가지고 있다면 그것을 적어 주면 됩니다.

만약 클라이언트에 보낼 서버 이름이 다를 경우 호스트의 실제 이름이 아닌 WWW를 사용하게 할 수도 있습니다.(IP를 입력해도 됩니다.) 만일 DNS가 존재하지 않는 것이라면 개인 홈페이지 서비스에도 영향을 줍니다. 그러한 이유 때문에 이 설정은 유의해야 됩니다.

#DocumentRoot - 기본적인 html 또는 php 등등의 관련 문서의 기본 디렉토리를 설정 합니다.

(심볼릭 링크 또는 앨리어스를 사용해 다른 위치를 적어 줄 수도 있습니다.)

<TIP>——————————————————————————-

각 디렉토리별 사용자 인증 설정하기

다음에 설명할 디렉토리 설정과 관련되어 이 부분을 추가 해 주면 됩니다.

```
AuthType Basic                                   - 사용자 인증 방법
AuthName cry                                     - 인증할 사용자명
```

그림 104: 그림. 6-22 ServerName 설정

그림 105: 그림. 6-23 DocumentRoot 설정

그림 106: 그림. 6-24 Directory 옵션 설정

```
AuthUserFile /usr/local/httpd/htdocs/.htpasswd    - 패스워드 파일의 위치
AuthGroupFile /dev/null          - 사용 그룹을 표시할 파일의 경로
```

패스워드 파일을 만들려면 아래와 같이 합니다.

```
/usr/local/apache/bin/htpasswd 파일을 사용하는 방법과 직접 작성하는 방법이 있습니다.
bin#.htpasswd -c 패스워드파일명 유저명
위와 같이 실행하면 새로운 패스워드 입력란이 나오고 거기에 패스워드를 입력하면 됩니다.
그리고 생성된 파일을 인증할 디렉토리로 옮기면 됩니다.
```

————————————————————————————————-

#Directory 옵션 - 각각의 디렉토리에 관한 권한 등의 여러 가지 설정을 하는 곳입니다.

<Directory "디렉토리 path">   </Directory>로 설정합니다.

여기에는 여러 가지 옵션이 주어지는데 차례로 보겠습니다.

옵션의 사용

```
Option 옵션명 옵션명 … 입니다.
```

옵션 종류

그림 107: 그림. 6-25 UserDir 설정

- None - 옵션을 주지 않습니다.
- All - 모든 옵션을 줍니다.
- Indexes - 해당 디렉토리 안에 DirectoryIndex 지시자 부분의 파일이 없을 경우 디렉토리와 파일들을 목록화해서 보여 줍니다.
- Includes - Server Side Includes를 가능하게 합니다.
- FollowSymLinks - 심볼릭 링크로의 이동을 가능하게 합니다.
- ExecCGI - CGI 풀그램을 실행 가능하게 합니다.
- AllowOverride 옵션명 - None일 경우 .htaccess 내용이 access.conf의 옵션에 의해 효력이 없어 집니다.(해당 디렉토리 안에 .htaccess 파일이 존재할 경우)

  즉, access.conf 설정의 어느 부분을 무시 할건지의 여부를 설정합니다.
- order 옵션명 - 해당 디렉토리에 대한 IP , DNS 필터링의 순서를 설정합니다.

#UserDir - 개인 홈 페이지 서비스를 위한 개인 홈페이지 디렉토리 설정 부분입니다.

그림 108: 그림. 6-26 UserDir 설정

즉, 개인의 디렉토리에 public_html 디렉토리를 만들면 홈페이지 서비스를 받을 수 있습니다. Ex>http://skycry.ath.cx/ userid 위의 디폴트 디렉토리 말고 다른 디렉토리를 추가 함으로서 여러 개의 디렉토리 서비스를 할 수 있습니다.

#UserDir 옵션 - 개인 홈페이지 디렉토리에 대한 접근 권한 등을 설정 합니다.

<Limit> </Limit> 로 각각의 접근을 제어 할 수도 있습니다. 옵션들은 앞 부분에서 설명한 대로 하면 됩니다. 물론 사용자 인증도 할 수 있습니다.

#DirectoryIndex - 브라우저가 아파치 서버로 처음 접근 했을 때 보여줄 문서를 설정합니다.

한 개 이상 나열이 가능하고, 앞에 설명한 디렉토리 설정에서 Option indexes로 되어 있고, DirectoryIndex 지시자에 표시한 파일이 디렉토리에 없을 경우 해당 디렉토리와 파일들은 목록화 되어 브라우저에 나타납니다.

#AccessFileName - 각각의 디렉토리의 접근을 제어하는 정보를 가진 파일 이름 입니다.

즉, 이 파일을 제어 하려는 디렉토리에 넣어 사용 합니다. 앞의 디렉토리 지시자와 연관하여 참고 하세요.

#.htaccess 파일에의 접근 - <Files 파일명> </Files>로 설정을 하며, 사용 되는 옵션은앞에서 설명한 것과 같습니다.

그리고 파일명을 .htaccess 가 아닌 다른 이름으로 앞의 AccessFileName에서 바꾸었다면 그 이름을 적어줘야 됩니다.

그림 109: 그림. 6-27 DirectoryIndex 설정

그림 110: 그림. 6-28 AccessFileName 설정

그림 111: 그림. 6-29 .htaccess 설정

그림 112: 그림. 6-30 CacheNegotiatedDocs 설정

그림 113: 그림. 6-31 UseCanonicalName 설정

#CacheNegotiatedDocs - 아파큥칠캭캭쳉”羞崵岾막曠下捉획®서에 대해서는 <Pragma: no-cache> 내용을 보냅니다.

즉, 프락시 서버들이 문서를 캐쉬하지 못하게 요청 합니다. # 를 제거하고 옵션을 실행 하면 모든 프락시가 문서를 캐쉬할 수 있게 합니다.

#UseCanonicalName - 아파치 1.3 버전에서 새롭게 등장한 기능이고, 이 옵션이 켜져 있을 경우

자기 참조 URL(응답이 오는 서버를 다시 가리키는 URL)을 만들 필요가 있을 때마다 공식적인 이름을 만들기 위해 ServerName와 Port를 사용 합니다. off 일 경우는 아파치는 가능할 경우 클라이언트가 제공하는 hostname:port를 사용 합니다. 이것은 또한 CGI 스크립트의 SERVER_NAME와 SERVER_PORT에도 영향을 줍니다.

#TypeConfig - mime.type 파일의 위치를 설정합니다.

#DefaultType - 서버에서 사용되는 문서중 MIME 타입으로 알 수 없는 문서에 대해 적용할 기본 MIME 타입을 설정합니다.

디폴트 값은 text/plain 이며 이 값은 주로 서버에 문서 파일이 많을 경우에 사용 합니다. 만일 바이너리나 실행 파일이 많을 경우라면 이 옵션은 application/octet-stream으로 적어 주면 됩니다. (웹 브라우저가 실행 파일 등을 텍스트 문서로 착각하고 브라우저 상에 표시하지 않기 위함 입니다.)

#mod_mine_magic - mod_mine_magic 모듈을 사용할 경우 해당 파일의 내용으로 파일 타입 힌트를 얻는 역

그림 114: 그림. 6-32 TypeConfig 설정

그림 115: 그림. 6-33 DefaultType 설정

그림 116: 그림. 6-34 mod_mine_magic 설정

그림 117: 그림. 6-35 HostnameLookups 설정

할을 합니다.

이것을 사용하기 위해서는 해당 모듈을 탑재 해야 합니다. <IfModule> </IfModule> 지시자는 이 모듈이 있을 때 가능합니다.

#HostnameLookups - 아파치는 클라이언트의 요청을 받아 들이면서 IP를 확인하고 확인한 주소를 DNS 서버에 다시 의뢰 하는 작업을 하기 때문에 상당한 시간을 요하게 됩니다. 디폴트 값인 off로 두고 사용하길 권장 합니다.

#ErrorLog - 에러를 기록할 파일의 위치를 적어 줍니다. <VirtualHost> 지시자에서 ErrorLog 설정을 하지 않았다면 여기서의 설정을을 따릅니다. 로그 파일의 경우는 계속 용량이 증가하기 때문에 수시로 용량을 줄여야 합니다.

#LogLevel - 로그메세지의 크기를 제어 하는 부분 입니다.

옵션에는 debug , info , notice , warn , error , crit ,alert , emerg 등이 있습니다.

#LogFormat - CustomLog 지시자에서 사용할 별명을 정의 합니다.

```
%h - 호스트명
%l - 리모트 로그 이름
%u - 사용자 인증에 사용된 유저명
%t - 시간
```

그림 118: 그림. 9-36 ErrorLog 설정

그림 119: 그림. 6-37 LogLevel 설정

그림 120: 그림. 6-38 LogFormat 설정

그림 121: 그림. 6-39 CustomLog 설정

```
%r - 요청한 내용의 첫번째 줄
%s - 서버 상태
%b - 전송량(헤더 포함)
%{헤더} - 요구된 헤더 내용
%U - 요구한 URL
```

#CustomLog - access_log 파일의 위치를 설정 합니다. <VirtualHost> 지시자에서 이 설정 부분이 없을 경우 여기서 설정한 위치로 모두 저장됩니다.

가상 호스트별로 따로 접근 로그 파일을 만들고 싶으면 이 설정을 각각의 <VirtualHost> 지시자 안에 넣어 줘야 합니다.

#agent_log , referrer_log - 만약 여러분들이 agent_log , referrer_log 파일을 생성 하고자 할 경우 #를 제거합니다.

#single logfile - 만약 여러분들이 access , agent , referrer 정보를 하나의 파일에 모두 저장 하기를 원한다면 combined 옵션을 이용 합니다.

#ServerSignature - 아파치 서버가 생성하는 페이지(에러문서 , FTP 디렉토리 list , mod_status , mod_info output , CGI 생성문서 제외)에 서버 버전과 가상 호스트 이름을 포함하는 행을 넣는 것을 설정합니다.

옵션에는 On | Off | Email 등이 있고 Email의 경우 ServerAdmin으로의 mailto: 링크도 포함 됩니다.

그림 122: 그림. 6-40 agent_log , referrer_log 설정

그림 123: 그림. 6-41 single logfile 설정

그림 124: 그림. 6-42 ServerSignature 설정

그림 125: 그림. 6-43 Alias 설정