HandBook

FreeBSD 공식 핸드북을 번역해 올립니다

공식 사이트의 업데이트와 다국어화에 조금 불만이 생겨서 그냥 여기다 정리해 올리고 있어요 .

- 파트 I. 시작하기
  - Chapter 1. 소개
  - Chapter 2. FreeBSD 설치하기
  - Chapter 3. FreeBSD 기초
- 파트 II. 일반적인 작업
- 파트 III. 시스템 관리
- 파트 IV. 네트워크 통신
  - Chapter 34. 고급 네트워킹
- 파트 V. 부록
- FreeBSD 과거 핸드북 번역판

□□ I. □□□

Chapter 1. 소개

1.1. 요약

FreeBSD에 관심을 가져주셔서 감사합니다 ! 다음 장에서는 FreeBSD 프로젝트의 역사 , 목표 , 개발 모델 등 다양한 측면을 다룹니다 .
이 장을 읽고 나면 다음 사항을 알게 될 것입니다 :

- FreeBSD와 다른 컴퓨터 운영체제의 관계 .
- FreeBSD 프로젝트의 역사 .
- FreeBSD 프로젝트의 목표 .
- FreeBSD 오픈 소스 개발 모델의 기본 사항 .
- 그리고 물론 "FreeBSD"라는 이름의 유래 .

1.2. FreeBSD에 오신 것을 환영합니다 !

FreeBSD는 x86(32비트 및 64비트 모두), ARM, AArch64, RISC-V, POWER 및 PowerPC 컴퓨터용 오픈 소스 표준 준수 유닉스 계열 운영체제입니다 . 선제적 멀티태스킹 , 메모리 보호 , 가상 메모리 , 다중 사용자 기능 , SMP 지원 , 다양한 언어와 프레임워크를 위한 모든 오픈 소스 개발 도구 , X Window System, KDE 또는 GNOME을 중심으로 한 데스크톱 기능 등 오늘날 당연하게 여겨지는 모든 기능을 제공합니다 . 특히 다음과 같은 강점이 있습니다 :

- 자유 오픈 소스 라이선스로 , 카피레프트 라이선스 특유의 제한 없이 소스 코드를 자유롭게 수정 및 확장하고 오픈 소스 프로젝트와 폐쇄형 제품 모두에 통합할 수 있는 권한을 부여하며 잠재적인 라이선스 비호환성 문제를 피할 수 있습니다 .
- 강력한 TCP/IP 네트워킹 - FreeBSD는 성능과 확장성이 계속 향상되고 있는 업계 표준 프로토콜을 구현합니다 . 따라서 서버와 라우팅 /방화벽 역할 모두에 적합하며 , 실제로 많은 기업과 공급업체가 이러한 목적으로 사용하고 있습니다 .

- root-on-ZFS, ZFS , , jails , FreeBSD , OpenZFS .
- Capsicum 30,000 API - FreeBSD FreeBSD "" .
- Linux Linux .

FreeBSD (CSRG) 4.4BSD-Lite , BSD . CSRG , FreeBSD . FreeBSD .

1.2.1. FreeBSD ?

FreeBSD , , UNIX® FreeBSD ! FreeBSD .

FreeBSD , FreeBSD :

- : FreeBSD TCP/IP :
  - Web servers
  - IPv4 IPv6 routing
  - Firewalls and NAT ("IP masquerading") gateways
  - FTP servers
  - Email servers
  - Storage servers

    - Virtualization servers
    - 이외 여러가지 ...
- 교육 : 컴퓨터 과학 또는 관련 공학 분야의 학생인가요 ? 운영 체제 , 컴퓨터 아키텍처 및 네트워킹에 대해 배울 수 있는 가장 좋은 방법은 FreeBSD가 제공할 수 있는 실무 경험보다 더 좋은 방법은 없습니다 . 무료로 제공되는 수많은 CAD, 수학 및 그래픽 디자인 패키지는 컴퓨터에 대한 주된 관심이 다른 작업을 수행하는 데 있는 사람들에게도 매우 유용합니다 !
- 연구 : 전체 시스템의 소스 코드를 사용할 수 있는 FreeBSD는 운영 체제는 물론 컴퓨터 과학의 다른 분야 연구를 위한 훌륭한 플랫폼입니다 . 또한 FreeBSD는 자유롭게 사용할 수 있기 때문에 특별한 라이선스 계약이나 공개 포럼에서 논의할 수 있는 내용에 대한 제한에 대해 걱정할 필요 없이 원격 그룹이 아이디어나 공동 개발을 위해 협업할 수 있습니다 .
- 네트워킹 : 새 라우터가 필요하신가요 ? 네임 서버 (DNS)가 필요하신가요 ? 내부 네트워크에서 사람들을 차단하기 위한 방화벽이 필요하신가요 ? FreeBSD는 구석에 방치되어 있는 사용하지 않는 PC를 정교한 패킷 필터링 기능을 갖춘 고급 라우터로 쉽게 전환할 수 있습니다 .
  임베디드 : FreeBSD는 임베디드 시스템을 구축하기 위한 훌륭한 플랫폼입니다 . 강력한 네트워크 스택 , 최첨단 기능 및 허용되는 BSD 라이선스와 함께 ARM, AArch64 및 PowerPC 플랫폼을 지원하는 FreeBSD는 임베디드 라우터 , 방화벽 및 기타 장치를 구축하기 위한 훌륭한 기반이 됩니다 .
- 데스크톱 : FreeBSD는 무료로 제공되는 X11 서버와 Wayland 디스플레이 서버를 사용하는 저렴한 데스크톱 솔루션을 위한 훌륭한 선택입니다 . FreeBSD는 표준 GNOME 및 KDE 그래픽 사용자 인터페이스를 포함한 많은 오픈 소스 데스크탑 환경 중에서 선택할 수 있습니다 . FreeBSD는 중앙 서버에서 "디스크 없이 " 부팅할 수도 있으므로 개별 워크스테이션을 더욱 저렴하고 쉽게 관리할 수 있습니다 .
- 소프트웨어 개발 : 기본 FreeBSD 시스템에는 완전한 C/C++ 컴파일러 및 디버거 제품군을 포함한 전체 개발 도구 세트가 함께 제공됩니다 . 포트 및 패키지 컬렉션을 통해 다른 많은 언어에 대한 지원도 제공됩니다 .

FreeBSD는 무료로 다운로드하거나 CD-ROM 또는 DVD로 구할 수 있습니다 . FreeBSD를 구하는 방법에 대한 자세한 내용은 FreeBSD 구하기 를 참조하세요 .

1.2.2. 누가 FreeBSD를 사용하나요 ?

FreeBSD는 웹 서비스 기능으로 잘 알려져 있습니다 . FreeBSD를 기반으로 제품과 서비스를 제공하는 회사들의 추천사 목록 은 FreeBSD 재단 웹사이트에서 확인할 수 있습니다 . 위키피디아에서도 FreeBSD 기반 제품 목록 을 관리하고 있습니다 .

1.3. FreeBSD 프로젝트 소개

다음 섹션에서는 프로젝트의 간략한 연혁 , 프로젝트 목표 , 개발___모델__ 등 프로젝트에 대한 몇 가지 배경 정보를 제공합니다 .

1.3.1. FreeBSD의 간략한 역사

FreeBSD 프로젝트는 1993년 초에 시작되었으며 , 부분적으로는 비공식 386BSDPatchkit의 마지막 코디네이터 3명의 아이디어로 시작되었습니다 : 네이트 윌리엄스 , 로드 그라임스 , 조던 허바드 .

원래 목표는 패치킷 메커니즘으로 해결할 수 없는 여러 문제를 해결하기 위해 386BSD의 중간 스냅샷을 생성하는 것이었습니다 . 이 프로젝트의 초기 작업 제목은 386BSD 0.5 또는 386BSD Interim이었습니다 .

386BSD는 빌 졸리츠의 운영체제였는데 , 그 당시까지 거의 1년 동안 방치되어 심각한 문제를 겪고 있었습니다 . 날이 갈수록 패치킷이 점점 더 불편하게 부풀어 오르자 , 그들은 임시 "정리 " 스냅샷을 제공하여 Bill을 돕기로 결정했습니다 . 이러한 계획은 빌 졸리츠가 갑자기 프로젝트에 대한 제재를 철회하기로 결정하고 대신 무엇을 할 것인지에 대한 명확한 언급 없이 무례하게 중단되었습니다 .

세 사람은 빌의 지원이 없어도 목표가 여전히 가치가 있다고 생각했고 , 데이비드 그린맨이 만든 "FreeBSD" 라는 이름을 채택했습니다 . 초기 목표는 현재 시스템 사용자들과 상의한 후 설정되었고 , 프로젝트가 현실화될 수 있는 단계에 이르렀다는 것이 분명해지자 조던은 인터넷에 쉽게 접근할 수 없는 많은 불행한 사람들을 위해 FreeBSD의 배포 채널을 개선하기 위해 Walnut Creek CDROM에 연락을 취했습니다 . 윌넛 크릭 CD롬은 FreeBSD를 CD로 배포하는 아이디어를 지지했을 뿐만 아니라 , 프로젝트에 작업할 컴퓨터와 빠른 인터넷 연결까지 제공했습니다 . 당시 완전히 알려지지 않은 프로젝트에 대한 Walnut Creek CDROM의 거의 전례 없는 믿음이 없었다면 , FreeBSD가 지금처럼 빠르게 , 그리고 멀리 퍼져나가지 못했을 것입니다 .

최초의 CD-ROM(및 일반 네트워크 전체) 배포는 1993년 12월에 출시된 FreeBSD 1.0이었습니다 . 이것은 U.C. 버클리의 4.3BSD-Lite("Net/2") 테이프를 기반으로 했으며 , 386BSD와 자유 소프트웨어 재단에서도 많은 구성 요소를 제공했습니다 . 첫 번째 제품치고는 꽤 괜찮은 성공을 거두었고 , 1994년 5월에는 매우 성공적인 FreeBSD 1.1을 출시했습니다 .

이 무렵 , 노벨과 UC 버클리가 버클리 Net/2 테이프의 법적 지위에 대한 오랜 소송을 합의하면서 다소 예상치 못한 폭풍우가 몰아쳤습니다 . 그 합의의 조건은 U.C. 버클리가 Net/2의 파일 3개가 "방해되는 " 코드이며 , 얼마 전에 AT&T로부터 인수한 Novel의 자산이므로 제거해야 한다는 것을 인정하는 것이었습니다 . 버클리가 그 대가로 얻은 것은 4.4BSD-Lite 릴리스가 마침내 출시될 때 방해받지 않는다고 선언하고 모든 기존 Net/2 사용자에게 전환을 강력히 권장하겠다는 Novell의 '축복 '이었습니다 . 여기에는 FreeBSD도 포함되었고 , 프로젝트는 1994년 7월 말까지 자체 Net/2 기반 제품의 출시를 중단할 수 있는 시간을 부여받았습니다 . 그 계약 조건에 따라 , 프로젝트는 마감일 전에 마지막 릴리스를 한 번 더 배포할 수 있었는데 , 그 릴리스가 바로 FreeBSD 1.1.5.1이었습니다 .

그 후 FreeBSD는 완전히 새롭고 다소 불완전한 4.4BSD-Lite 비트 세트에서 말 그대로 재창조하는 고된 작업에 착수했습니다 . 시스템 V 공유 메모리 및 세마포어와 관련된 파일 3개만 제거되었지만 , 그 외 많은 변경 사항과 버그 수정이 BSD 배포판에 이루어졌기 때문에 모든 FreeBSD 개발 사항을 4.4BSD-Lite에 통합하는 것은 엄청난 작업이었습니다 . 프로젝트는 1994년 11월까지 이 전환 작업을 진행했고 , 12월에 FreeBSD 2.0 을 세상에 공개했습니다 . 이 릴리스는 여전히 가장자리가 약간 거칠긴 했지만 큰 성공을 거두었고 , 1995년 6

월에 더 강력하고 설치하기 쉬운 FreeBSD 2.0.5 릴리스가 이어졌습니다 .

그 이후로 FreeBSD는 매번 이전 버전의 안정성 , 속도 및 기능 세트를 개선하는 일련의 릴리스를 만들었습니다 .

현재 장기 개발 프로젝트는 15.0-CURRENT(메인) 브랜치에서 계속 진행 중이며 , 작업이 진행됨에 따라 15.0 의 스냅샷 릴리스가 스냅샷 서버 에서 지속적으로 제공됩니다 .

1.3.2. FreeBSD 프로젝트 목표

FreeBSD 프로젝트의 목표는 어떤 용도로든 아무런 조건 없이 사용할 수 있는 소프트웨어를 제공하는 것입니다 . 우리 중 많은 사람들이 코드 (와 프로젝트)에 상당한 투자를 하고 있으며 , 때때로 약간의 금전적 보상을 받는 것도 마다하지 않겠지만 , 우리는 그것을 고집할 준비가 되어 있지 않습니다 . 저희의 가장 중요한 ' 사명 '은 어떤 사람이든 , 어떤 목적으로든 코드를 제공함으로써 코드가 최대한 널리 사용되고 최대한 많은 혜택을 제공하도록 하는 것이라고 믿습니다 . 이것이 바로 자유 소프트웨어의 가장 근본적인 목표 중 하나이자 우리가 열렬히 지지하는 목표라고 생각합니다 .

소스 트리에 있는 코드 중 GNU 일반 공중 라이선스 (GPL) 또는 라이브러리 일반 공중 라이선스 (LGPL)에 해당하는 코드에는 일반적인 경우보다는 강제 액세스 측면에서 약간 더 많은 제약이 따릅니다 . 그러나 GPL 소프트웨어의 상업적 사용에서 발생할 수 있는 추가적인 복잡성 때문에 , 저희는 합리적인 선택이라면 보다 완화된 BSD 라이선스에 따라 제출된 소프트웨어를 선호합니다 .

1.3.3. FreeBSD 개발 모델

FreeBSD의 개발은 매우 개방적이고 유연한 과정 이며 , 기여자 목록 에서 볼 수 있듯이 말 그대로 전 세계 수천 명의 사람들의 기여를 통해 구축되고 있습니다 . FreeBSD의 개발 인프라는 이 수천 명의 기여자들이 인터넷을 통해 협업할 수 있도록 해줍니다 . 저희는 지속적으로 새로운 자원 봉사자를 찾고 있으며 , 더 긴밀하게 참여하고 싶은 분들은 FreeBSD에 기여하기 문서를 참조하시기 바랍니다 .

독립적으로 작업하든 긴밀하게 협력하든 FreeBSD 프로젝트와 그 개발 과정에 대해 알아두면 유용한 정보입니다 :

Git 저장소

몇 년 동안 , FreeBSD의 중앙 소스 트리는 무료로 제공되는 소스 코드 제어 도구인 CVS(Concurrent Versions System)에 의해 유지 관리되었습니다 . 2008년 6월 , 프로젝트는 SVN(Subversion)을 사용하기로 전환했습니다 . 소스 트리의 급속한 확장과 이미 저장된 기록의 양으로 인해 CVS의 기술적 한계가 분명해졌기 때문에 전환이 필요하다고 판단되었습니다 . 문서화 프로젝트와 포트 컬렉션

리포지토리도 각각 2012년 5월과 2012년 7월에 CVS에서 SVN으로 이전했습니다 . 2020년 12월에 프로젝트는 소스 및 문서 리포지토리를 Git으로 마이그레이션 했고 , 2021년 4월에 포트 도 그 뒤를 따랐습니다 . FreeBSD src/ 저장소를 얻는 방법에 대한 자세한 내용은 소스 얻기 섹션을 , FreeBSD 포트 컬렉션을 얻는 방법에 대한 자세한 내용은 포트 컬렉션 사용하기 섹션을 참고하세요 .

커미터 목록

커미터는 Git 저장소에 대한 푸시 액세스 권한을 가지고 있으며 , FreeBSD 소스를 수정할 권한이 있는 사람들입니다 (커미터라는 용어는 저장소에 새로운 변경 사항을 가져오는 데 사용되는 소스 제어 명령어인 커밋에서 유래했습니다). 누구나 버그 데이터베이스에 버그를 제출할 수 있습니다 . 버그 리포트를 제출하기 전에 FreeBSD 메일링 리스트 , IRC 채널 또는 포럼을 사용하여 문제가 실제로 버그인지 확인하는 데 도움을 받을 수 있습니다 .

FreeBSD 코어 팀

FreeBSD 코어 팀은 FreeBSD 프로젝트가 회사라면 이사회에 해당할 것입니다 . 코어 팀의 주요 임무는 프로젝트가 전체적으로 양호한 상태이며 올바른 방향으로 나아가고 있는지 확인하는 것입니다 . 헌신적이고 책임감 있는 개발자를 커미터 그룹에 초대하는 것도 코어 팀의 기능 중 하나이며 , 다른 코어 팀원들이 자리를 옮길 때 새로운 코어 팀원을 모집하는 것도 마찬가지입니다 . 현재 코어 팀은 2022년 5월에 커미터 후보자 풀에서 선출되었습니다 . 선거는 2년마다 실시됩니다 .

> **NOTE**
> 대부분의 개발자와 마찬가지로 , 핵심 팀의 구성원들 역시 FreeBSD 개발에 있어서 자원 봉사자이며 프로젝트의 재정적 혜택을 받지 않기 때문에 "헌신 "을 "지원 보장 "의 의미로 오해해서는 안 됩니다 . 위의 "이사회 "라는 비유는 그다지 정확하지 않으며 , 이들은 자신의 더 나은 판단에 반하여 FreeBSD를 위해 자신의 삶을 포기한 사람들이라고 말하는 것이 더 적합할지도 모릅니다 !

FreeBSD 재단

FreeBSD 재단 은 501(c)(3)의 미국 기반 비영리 단체로 , 전 세계 FreeBSD 프로젝트와 커뮤니티를 지원하고 홍보하는 데 전념하고 있습니다 . 재단은 프로젝트 보조금을 통해 소프트웨어 개발에 자금을 지원하고 긴급한 문제에 즉시 대응하고 새로운 기능을 구현할 수 있는 직원을 제공합니다 . 재단은 FreeBSD 인프라를 개선하고 유지하기 위해 하드웨어를 구입하고 테스트 커버리지 , 지속적인 통합 및 자동화를 개선하기 위해 인건비를 지원합니다 . 재단은 전 세계의 기술 컨퍼런스 및 행사에서 FreeBSD를 홍보함으로써 FreeBSD를 지지합니다 . 재단은 또한 더 많은 사용자와 기여자를 모집하기 위해 워크샵 , 교육 자료 , 프레젠테이션을 제공합니다 . 또한 , 재단은 공인된 법인을 필요로 하는 계약 , 라이선스 계약 및 기타 법적 약정을 체결할 때 FreeBSD 프로젝트를 대표합니다 .

외부 기여자

마지막으로 , 가장 큰 개발자 그룹은 거의 지속적으로 피드백과 버그 수정을 제공하는 사용자들입니다 . FreeBSD 기본 시스템의 개발에 대한 최신 소식을 접할 수 있는 가장 좋은 방법은 이러한 사항들이

논의되는 FreeBSD 기술 토론 메일링 리스트에 가입하는 것입니다. 타사 애플리케이션 포팅의 경우, FreeBSD 포트 메일링 리스트입니다. 문서의 경우 - FreeBSD 문서 프로젝트 메일링 리스트. 다양한 FreeBSD 메일링 리스트에 대한 자세한 정보는 인터넷의 리소스를 참고하세요.

FreeBSD 기여자 목록은 길고 성장하고 있는 목록이므로, 오늘 FreeBSD에 무언가를 기여 함으로써 참여해보시는 것은 어떨까요? 코드를 제공하는 것만이 유일한 방법은 아닙니다!

요약하자면, 우리의 개발 모델은 느슨한 동심원 집합으로 구성되어 있습니다. 중앙 집중식 모델은 잠재적인 기여자를 배제하기 위한 것이 아니라, 하나의 중앙 코드 베이스를 쉽게 추적할 수 있는 방법을 제공하는 FreeBSD 사용자들의 편의를 위해 설계되었습니다! 우리의 바람은 사용자가 쉽게 설치하고 사용할 수 있는 일관된 대규모 응용 프로그램 세트를 갖춘 안정적인 운영 체제를 제공하는 것이며, 이 모델은 이를 달성하는 데 매우 효과적입니다.

FreeBSD의 개발자로 참여하고자 하는 분들에게 바라는 것은 현재의 사람들이 지속적인 성공을 위해 헌신하는 것뿐입니다!

1.3.4. 서드파티 프로그램

기본 배포판 외에도 FreeBSD는 일반적으로 많이 찾는 수천 개의 프로그램이 포함된 포팅된 소프트웨어 모음을 제공합니다. 포트 목록은 HTTP 서버부터 게임, 언어, 편집기, 그리고 그 사이의 거의 모든 것에 이르기까지 다양합니다. 약 36,000개의 포트가 있으며, 전체 포트 컬렉션에는 약 3GB가 필요합니다. 포트를 컴파일하려면 설치하려는 프로그램의 디렉터리로 변경하고 make install을 입력하기만 하면 나머지는 시스템이 알아서 처리합니다. 빌드하는 각 포트의 전체 원본 배포는 동적으로 검색되므로 원하는 포트를 빌드하는 데 충분한 디스크 공간만 있으면 됩니다.

거의 모든 포트는 미리 컴파일된 "패키지"로도 제공되므로 소스에서 직접 포트를 컴파일하고 싶지 않은 사용자는 간단한 명령 (pkg 설치)으로 설치할 수 있습니다. 패키지 및 포트에 대한 자세한 내용은 애플리케이션 설치하기에서 확인할 수 있습니다: 패키지 및 포트를 참조하세요.

1.3.5. 추가 문서

지원되는 모든 FreeBSD 버전은 초기 시스템 설정 시 설치 프로그램에서 **/usr/local/share/doc/freebsd** 아래에 추가 문서를 설치할 수 있는 옵션을 제공합니다. 나중에 패키지를 사용하여 문서를 설치할 수도 있습니다:

```
$ pkg install en-freebsd-doc
```

또는 현지화된 버전은 'en'을 원하는 언어 접두사로 대체합니다. 현지화된 버전 중 일부는 오래되어 더 이상 정확하지 않거나 관련성이 없는 정보가 포함되어 있을 수 있습니다. 다음 URL을 사용하여 웹 브라우저에서

로컬로 설치된 매뉴얼을 볼 수 있습니다 :

FreeBSD 핸드북

/usr/local/share/doc/freebsd/en/books/handbook/handbook_en.pdf

FreeBSD FAQ

/usr/local/share/doc/freebsd/en/books/faq/faq_en.pdf

문서 포털에서 항상 최신 문서를 찾을 수 있습니다 .

모든 상표는 해당 소유자의 자산입니다 .

마지막 수정 날짜 : 2023 년 11 월 6 일

작성자 : Ceri Davies

Chapter 2. FreeBSD 설치하기

2.1. 요약

FreeBSD는 amd64, ARM®, RISC-V® 및 PowerPC®를 포함한 다양한 아키텍처를 지원합니다. 아키텍처와 플랫폼에 따라 다른 이미지를 다운로드하여 설치하거나 직접 실행할 수 있습니다.

이미지 유형은 다음과 같습니다:

- 가상 머신 디스크 이미지 (예: qcow2, vmdk, vhd 및 원시 장치 이미지). 이러한 이미지는 설치 이미지가 아니라 설치 후 작업에 사용할 수 있도록 FreeBSD가 사전 설치되어 있는 이미지입니다. 가상 머신 이미지는 클라우드 환경에서도 일반적으로 사용됩니다.
- SD 카드 이미지, 라즈베리 파이와 같은 임베디드 시스템용. 이러한 파일은 압축되지 않은 원시 이미지로 SD 카드에 기록되어야 하며, 이 이미지에서 보드가 부팅됩니다.
- 설치 이미지는 일반적인 데스크톱, 노트북 또는 서버 시스템용 드라이브에 FreeBSD를 설치하기 위해 ISO 또는 USB 장치에서 부팅할 때 사용합니다.

이 장의 나머지 부분에서는 세 번째 경우에 대해 설명하며, bsdinstall이라는 텍스트 기반 설치 프로그램을 사용하여 FreeBSD를 설치하는 방법을 설명합니다. 설치 프로그램과 여기에 표시된 것 사이에는 약간의 차이가 있을 수 있으므로 이 장은 문자 그대로의 지침이 아니라 일반적인 지침으로 사용하십시오.

이 장을 읽고 나면 다음 사항을 알게 될 것입니다:

- FreeBSD 이미지를 구하고 FreeBSD 설치 미디어를 만드는 방법.
- bsdinstall을 시작하는 방법.
- bsdinstall이 묻는 질문과 그 의미, 그리고 대답하는 방법.
- 설치 실패 문제를 해결하는 방법.
- 설치를 커밋하기 전에 FreeBSD의 라이브 버전에 액세스하는 방법.

2.2. 하드웨어 최소 요구사항

FreeBSD를 설치하기 위한 하드웨어 요구 사항은 아키텍처와 버전에 따라 다릅니다. FreeBSD 릴리스에서 지원하는 하드웨어 아키텍처 및 디바이스는 FreeBSD 릴리스 정보 페이지에 나와 있습니다. FreeBSD 다운로드 페이지에는 다양한 아키텍처에 적합한 이미지를 선택하기 위한 권장 사항도 있습니다.

2.3. 설치 전 작업

시스템이 FreeBSD 설치를 위한 최소 하드웨어 요구 사항을 충족한다고 판단되면, 설치 파일을 다운로드하고 설치 미디어를 준비해야 합니다. 이 작업을 수행하기 전에 이 체크리스트의 항목을 확인하여 시스템이 설치 준비가 되었는지 확인하세요:

1) 중요한 데이터 백업

운영 체제를 설치하기 전에 항상 모든 중요한 데이터를 먼저 백업하세요. 백업을 설치 중인 시스템에 저장하지 마세요. 대신 USB 드라이브, 네트워크의 다른 시스템 또는 온라인 백업 서비스와 같은 이동식 디스크에 데이터를 저장하세요. 설치를 시작하기 전에 백업을 테스트하여 필요한 파일이 모두 포함되어 있는지 확인합니다. 설치 프로그램이 시스템 디스크를 포맷하면 해당 디스크에 저장된 모든 데이터가 손실됩니다.

2) FreeBSD를 설치할 위치 결정하기

FreeBSD만 설치되는 운영 체제인 경우 이 단계를 건너뛸 수 있습니다. 하지만 다른 운영 체제와 디스크를 공유할 경우, FreeBSD에 사용할 디스크 또는 파티션을 결정하세요.

i386 및 amd64 아키텍처에서는 두 가지 파티셔닝 방식 중 하나를 사용하여 디스크를 여러 파티션으로 나눌 수 있습니다. 기존의 MBR(마스터 부트 레코드)은 최대 4개의 기본 파티션을 정의하는 파티션 테이블을 보유합니다. 역사적인 이유로, FreeBSD는 이러한 기본 파티션 슬라이스라고 부릅니다. 이러한 기본 파티션 중 하나는 여러 논리 파티션을 포함하는 확장 파티션으로 만들 수 있습니다. GUID 파티션 테이블 (GPT)은 보다 새롭고 간단한 디스크 파티션 방법입니다. 일반적인 GPT 구현은 디스크당 최대 128개의 파티션을 허용하므로 논리 파티션이 필요하지 않습니다.

FreeBSD 부트 로더에는 기본 파티션 또는 GPT 파티션이 필요합니다. 기본 파티션 또는 GPT 파티션이 모두 이미 사용 중이라면 FreeBSD를 위해 하나를 비워야 합니다. 기존 데이터를 삭제하지 않고 파티션을 생성하려면 파티션 크기 조정 도구를 사용하여 기존 파티션을 축소하고 확보된 공간을 사용하여 새 파티션을 생성합니다.

다양한 무료 및 상용 파티션 크기 조정 도구는 디스크_파티셔닝_소프트웨어_위키백과_항목 에 나열되어 있습니다. GParted Live는 GParted 파티션 편집기가 포함된 무료 라이브 CD입니다.

> 디스크 축소 유틸리티를 올바르게 사용하면 새 파티션을 만들기 위한 공간을 안전하게 만들 수 있습니다. 잘못된 파티션을 선택할 가능성이 있으므로 디스크 파티션을 수정하기 전에 항상 중요한 데이터를 백업하고 백업의 무결성을 확인하세요.

서로 다른 운영 체제가 포함된 디스크 파티션을 사용하면 한 컴퓨터에 여러 운영 체제를 설치할 수 있습니다. 디스크 파티션을 수정하지 않고도 여러 운영 체제를 동시에 실행할 수 있는 가상화를 사용하는 것도 대안이 될

수 있습니다 .

3) 네트워크 정보 수집

일부 FreeBSD 설치 방법은 설치 파일을 다운로드하기 위해 네트워크 연결이 필요합니다 . 설치 후 설치 관리자는 시스템의 네트워크 인터페이스를 설정하도록 제안합니다 .

네트워크에 DHCP 서버가 있는 경우 , 이 서버를 사용하여 자동 네트워크 구성을 제공할 수 있습니다 . DHCP를 사용할 수 없는 경우 로컬 네트워크 관리자 또는 인터넷 서비스 제공업체로부터 시스템에 대한 다음 네트워크 정보를 얻어야 합니다 :

필수 네트워크 정보

1. IP 주소
2. 서브넷 마스크
3. 기본 게이트웨이의 IP 주소
4. 네트워크의 도메인 이름
5. 네트워크의 DNS 서버의 IP 주소

4) FreeBSD 정오표 확인

FreeBSD 프로젝트는 FreeBSD의 각 릴리스가 가능한 한 안정적이 되도록 노력하지만 , 때때로 버그가 프로세스에 스며들기도 합니다 . 아주 드물게 이러한 버그가 설치 프로세스에 영향을 미치는 경우가 있습니다 . 이러한 문제가 발견되고 수정되면 각 버전의 FreeBSD 에라타 페이지에 기록됩니다 . 설치하기 전에 에라타를 확인하여 설치에 영향을 줄 수 있는 문제가 없는지 확인하세요 .

모든 릴리스에 대한 정보 및 오류는 FreeBSD 릴리스 정보 페이지 에서 확인할 수 있습니다 .

2.3.1. 설치 미디어 준비

FreeBSD 설치 프로그램은 다른 운영 체제 내에서 실행할 수 있는 애플리케이션이 아닙니다 . 대신 FreeBSD 설치 파일을 다운로드하여 해당 파일 유형 및 크기와 관련된 미디어 (CD, DVD 또는 USB)에 구운 다음 시스템을 부팅하여 삽입된 미디어에서 설치합니다 .

FreeBSD 설치 파일은 FreeBSD 다운로드 페이지 에서 다운로드할 수 있습니다 . 각 설치 파일의 이름에는 FreeBSD의 릴리스 버전 , 아키텍처 및 파일 유형이 포함되어 있습니다 .

설치 파일은 xz(1) 또는 비압축으로 압축된 여러 가지 형식으로 제공됩니다 . 형식은 컴퓨터 아키텍처와 미디어 유형에 따라 다릅니다 .

설치 파일 유형 :

- **-bootonly.iso**: 이 파일은 인스톨러만 포함하므로 가장 작은 설치 파일입니다. 설치 프로그램이 FreeBSD 설치를 완료하는 데 필요한 파일을 다운로드하므로 설치하는 동안 인터넷 연결이 필요합니다. 이 파일은 광 미디어에 구워야 합니다.
- **-disc1.iso**: 이 파일에는 FreeBSD, 소스 및 포트 컬렉션을 설치하는 데 필요한 모든 파일이 포함되어 있습니다. 이 파일은 광 미디어에 구워야 합니다.
- **-dvd1.iso**: 이 파일에는 FreeBSD, 소스 및 포트 컬렉션을 설치하는 데 필요한 모든 파일이 포함되어 있습니다. 또한 인터넷에 연결하지 않고도 미디어에서 전체 시스템을 설치할 수 있도록 윈도우 매니저와 일부 애플리케이션을 설치하기 위한 인기 있는 바이너리 패키지 세트가 포함되어 있습니다. 이 파일은 광 미디어에 구워야 합니다.
- **-memstick.img**: 이 파일에는 FreeBSD, 소스 및 포트 컬렉션을 설치하는 데 필요한 모든 파일이 포함되어 있습니다. USB에 이미지 파일 쓰기에 표시된 대로 이 파일을 USB 스틱에 씁니다.
- **-mini-memstick.img**: **-bootonly.iso**와 마찬가지로 설치 파일을 포함하지 않고 필요에 따라 다운로드합니다. 설치하는 동안 인터넷 연결이 필요합니다. 이미지 파일을 USB에 쓰기에 표시된 대로 USB 스틱에 기록해야 합니다.

이미지 파일을 다운로드한 후 같은 디렉터리에서 체크섬 파일을 하나 이상 다운로드합니다. 릴리스 번호와 아키텍처 이름을 따서 명명된 두 개의 체크섬 파일을 사용할 수 있습니다. 예를 들어 CHECKSUM.SHA256-FreeBSD-13.1-RELEASE-amd64 및 CHECKSUM.SHA512-FreeBSD-13.1-RELEASE-amd64.

파일 중 하나 (또는 두 파일 모두)를 다운로드한 후 이미지 파일의 체크섬을 계산하여 체크섬 파일에 표시된 것과 비교합니다. 계산된 체크섬은 서로 다른 두 가지 알고리즘에 해당하므로 올바른 파일과 비교해야 한다는 점에 유의하세요: SHA256(1)과 SHA512(2). FreeBSD는 체크섬을 계산하는 데 사용할 수 있는 sha256(1) 및 sha512(1)를 제공합니다. 다른 운영 체제에도 유사한 프로그램이 있습니다.

FreeBSD에서 체크섬을 확인하려면 sha256sum(1)(및 sha512sum(1))을 실행하여 자동으로 수행할 수 있습니다:

```
$ sha256sum -c CHECKSUM.SHA256-FreeBSD-13.1-RELEASE-amd64 FreeBSD-13.1-RELEASE-amd64-dvd1.iso
FreeBSD-13.1-RELEASE-amd64-dvd1.iso: OK
```

체크섬이 정확히 일치해야 합니다. 체크섬이 일치하지 않으면 이미지 파일이 손상된 것이므로 다시 다운로드해야 합니다.

2.3.1.1. USB에 이미지 파일 쓰기

***memstick.img** 파일은 메모리 스틱의 전체 내용을 담은 이미지 입니다. 이 파일은 대상 장치에 파일로 복사할 수 없습니다. USB 스틱에 *.img를 쓰는 데는 여러 가지 애플리케이션을 사용할 수 있습니다. 이 섹션에서는 이러한 유틸리티 중 두 가지에 대해 설명합니다.

계속 진행하기 전에 USB 스틱에 있는 중요한 데이터를 백업하세요. 이 절차를 수행하면 스틱의 기존 데이터가 지워집니다.

절차 . **dd**를 사용하여 이미지 쓰기

이 예에서는 이미지가 기록될 대상 디바이스로 `/dev/da0`을 사용합니다 . 이 명령은 지정된 대상 디바이스의 기존 데이터를 파괴하므로 올바른 디바이스 를 사용해야 합니다 .

1. 명령줄 유틸리티는 BSD, Linux® 및 Mac OS® 시스템에서 사용할 수 있습니다 . `dd`를 사용하여 이미지를 구우려면 USB 스틱을 삽입하고 해당 장치 이름을 확인합니다 . 그런 다음 다운로드한 설치 파일의 이름과 USB 스틱의 장치 이름을 지정합니다 . 이 예에서는 기존 FreeBSD 시스템의 첫 번째 USB 장치에 amd64 설치 이미지를 구웁니다 .

```
dd if=FreeBSD-13.1-RELEASE-amd64-memstick.img of=/dev/da0 bs=1M conv=sync
```

이 명령이 실패하면 USB 스틱이 마운트되지 않았는지 , 장치 이름이 파티션이 아닌 디스크용인지 확인하세요 .

일부 운영 체제에서는 이 명령을 sudo(8)과 함께 실행해야 할 수도 있습니다 . dd(1) 구문은 플랫폼마다 약간씩 다릅니다 (예 : Mac OS®에서는 소문자 `bs=1m`이 필요합니다). Linux®와 같은 시스템에서는 쓰기가 버퍼링될 수 있습니다 . 모든 쓰기를 강제로 완료하려면 sync(8)을 사용합니다 .

절차 . **Windows®**를 사용하여 이미지 쓰기

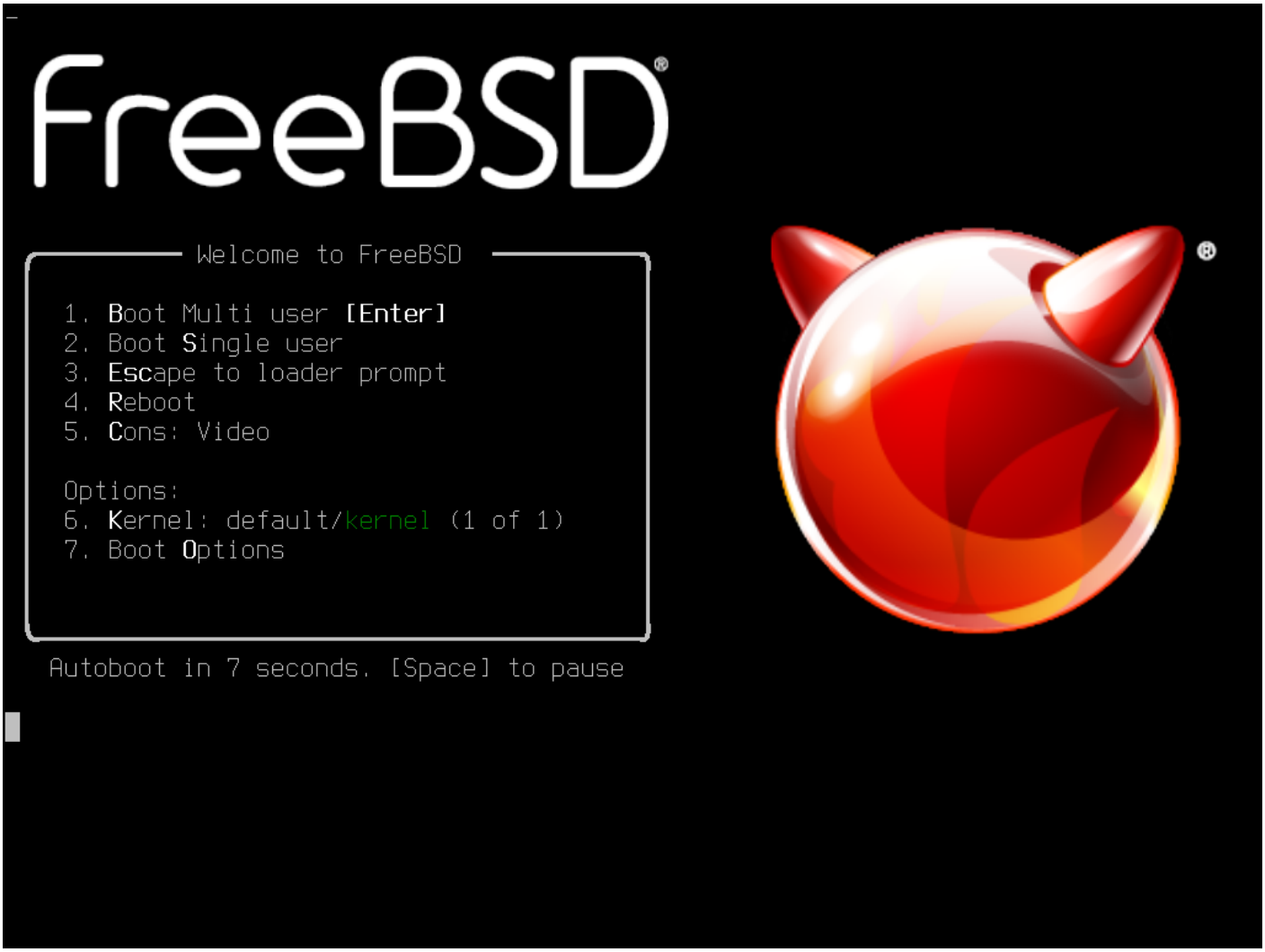
지정된 드라이브의 기존 데이터는 덮어쓰고 삭제되므로 올바른 드라이브 문자를 입력해야 합니다 .

1. Windows®용 Image Writer 받기
   Windows®용 Image Writer는 메모리 스틱에 이미지 파일을 올바르게 쓸 수 있는 무료 애플리케이션입니다 . win32diskimager 홈페이지에서 다운로드하여 폴더에 압축을 풉니다 .

2. 이미지 라이터로 이미지 쓰기
   Win32DiskImager 아이콘을 더블클릭하여 프로그램을 시작합니다 . Device 아래에 표시된 드라이브 문자가 메모리 스틱이 있는 드라이브인지 확인합니다 . 폴더 아이콘을 클릭하고 메모리 스틱에 기록할 이미지를 선택합니다 . **Save**를 클릭하여 이미지 파일 이름을 수락합니다 . 모든 것이 올바른지 , 메모리 스틱의 폴더가 다른 창에 열려 있지 않은지 확인합니다 . 모든 준비가 완료되면 **Write**를 클릭하여 메모리 스틱에 이미지 파일을 씁니다 .

2.4. 설치 시작하기

기본적으로 설치 시 다음 메시지 이전에는 디스크가 변경되지 않습니다 :



Your changes will now be written to disk. If you have chosen to overwrite existing data, it will be PERMANENTLY ERASED. Are you sure you want to commit your changes?

이 경고가 표시되기 전에 언제든지 설치를 종료할 수 있습니다 . 구성이 잘못되었다고 우려되는 경우 이 시점 전에 컴퓨터를 끄면 시스템 디스크가 변경되지 않습니다 .

이 섹션에서는 설치 미디어 준비하기의 지침에 따라 준비된 설치 미디어에서 시스템을 부팅하는 방법을 설명합니다 . 부팅 가능한 USB 스틱을 사용하는 경우 컴퓨터를 켜기 전에 USB 스틱을 연결합니다 . CD 또는 DVD로 부팅하는 경우 컴퓨터를 켜고 처음에 미디어를 삽입합니다 . 삽입된 미디어에서 부팅하도록 시스템을 구성하는 방법은 아키텍처에 따라 다릅니다 .

2.4.1. FreeBSD 부팅 메뉴

설치 미디어에서 시스템이 부팅되면 다음과 유사한 메뉴가 표시됩니다 :

Figure 1. FreeBSD Boot Loader Menu

기본적으로 메뉴는 사용자 입력을 10초간 기다린 후 FreeBSD 설치 프로그램으로 부팅하거나 , 이미 설치되어 있는 경우 FreeBSD로 부팅합니다 . 선택 항목을 검토하기 위해 부팅 타이머를 일시 중지하려면 스페이스키 를 누릅니다 . 옵션을 선택하려면 강조 표시된 숫자 , 문자 또는 키를 누릅니다 . 다음 옵션을 사용할 수 있습니다 .

- **Boot Multi User**: FreeBSD 부팅 프로세스를 계속 진행합니다 . 부팅 타이머가 일시 중지된 경우 **1** , 대문자 또는 소문자 **B**를 누르거나 **Enter** 키를 누릅니다 .

- **Boot Single User**: 이 모드는 "단일 사용자 모드 "에 설명된 대로 기존 FreeBSD 설치를 수정하는 데 사용할 수 있습니다 . 이 모드로 들어가려면 **2** 또는 대문자 또는 소문자 **S**를 누릅니다 .

- **Escape to loader prompt**: 이렇게 하면 제한된 수의 저수준 명령이 포함된 복구 프롬프트로 시스템이 부팅됩니다 . 이 프롬프트는 '3단계 '에 설명되어 있습니다 . 이 프롬프트로 부팅하려면 **3** 또는 **Esc** 키를 누릅니다 .

- **Reboot**: 시스템을 재부팅합니다 .
- **Cons**: video, serial, Dual (serial primary) 또는 Dual (Video primary)로 설치를 계속할 수 있도록 허용합니다 .

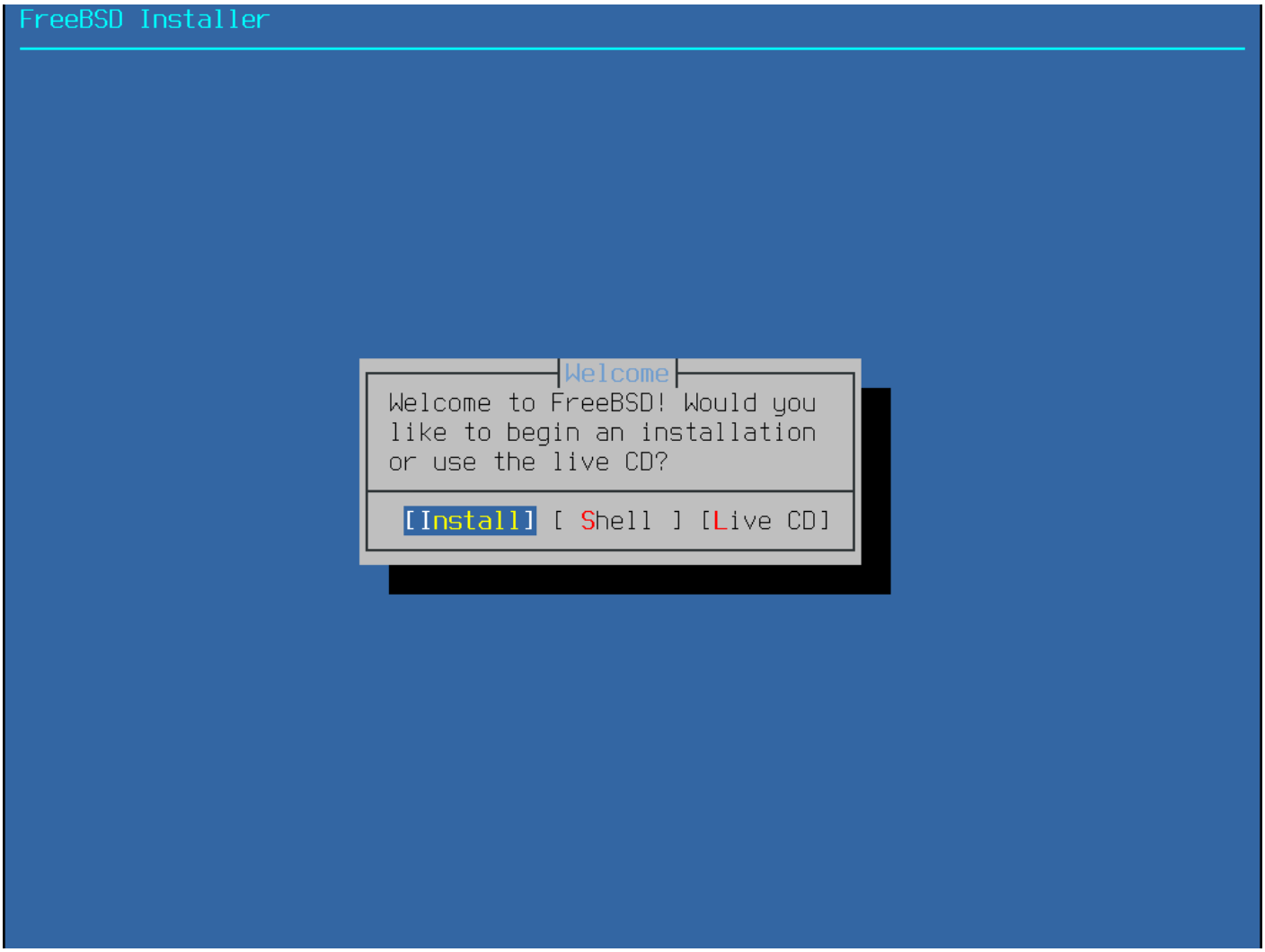
- **Kernel**: 다른 커널을 로드합니다 .

- **Boot Options**: FreeBSD 부팅 옵션 메뉴에 표시되고 아래에 설명된 메뉴를 엽니다 .

Figure 2. FreeBSD Boot Options Menu

On Off

- **ACPI Support**: Off ACPI
- **Safe Mode**: ACPI
- **Single User**: "" "" FreeBSD Off
- **Verbose**:

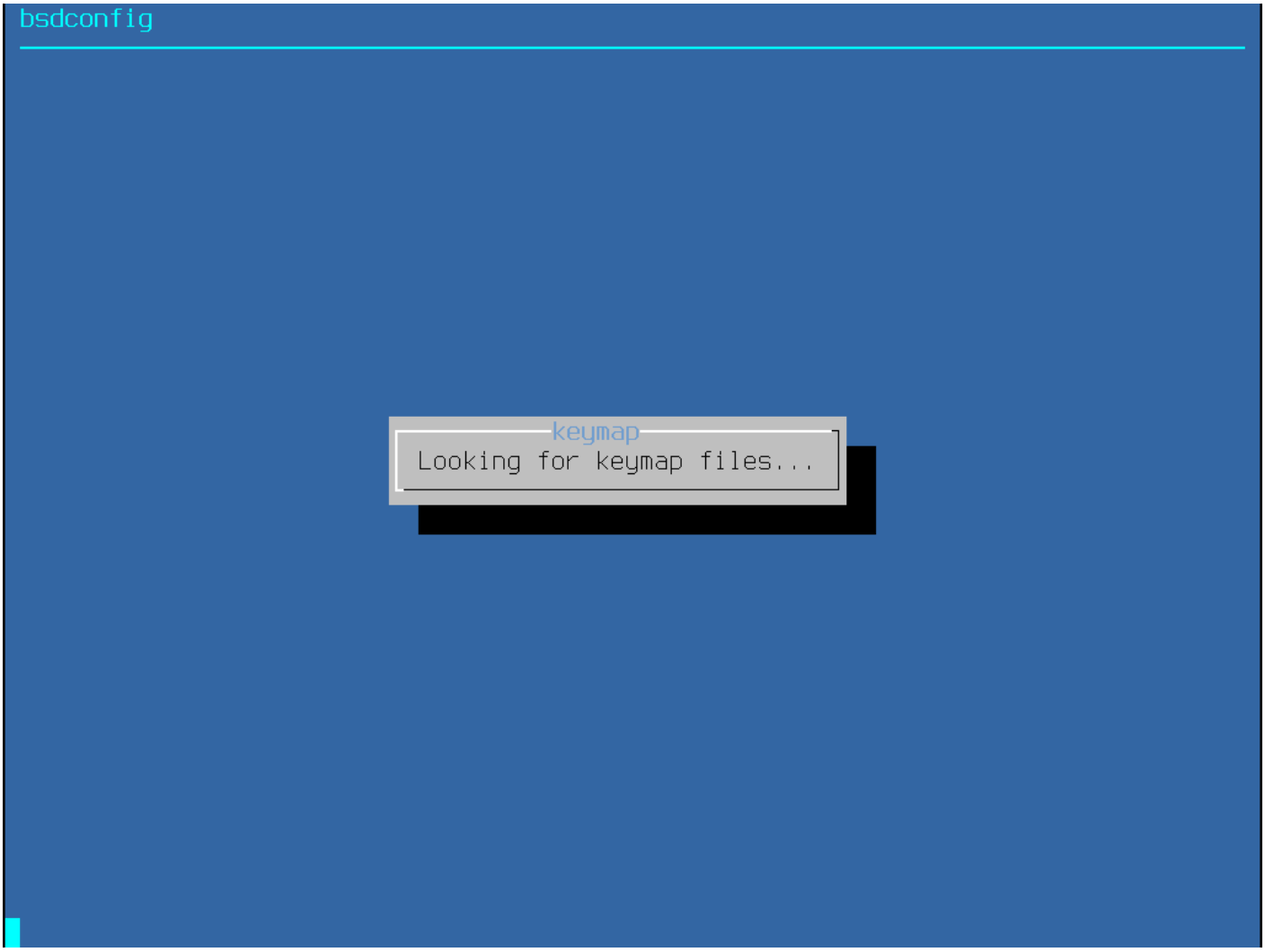


필요한 선택을 마친 후 **1** 또는 백스페이스 키를 눌러 기본 부팅 메뉴로 돌아간 다음 **Enter** 키를 눌러 FreeBSD로 부팅을 계속합니다. 일련의 부팅 메시지가 표시되면 FreeBSD가 하드웨어 장치 프로브를 수행하고 설치 프로그램을 로드합니다. 부팅이 완료되면 시작 메뉴에 표시된 시작 메뉴가 표시됩니다.

Figure 3. Welcome Menu

Enter 키를 눌러 기본값인 설치를 선택하여 설치 프로그램으로 들어갑니다. 이 장의 나머지 부분에서는 이 설치 프로그램을 사용하는 방법을 설명합니다. 그렇지 않으면 오른쪽 또는 왼쪽 화살표 또는 색상이 지정된 문자를 사용하여 원하는 메뉴 항목을 선택합니다. **Shell**은 설치 전에 명령줄 유틸리티를 사용하여 디스크를 준비하기 위해 FreeBSD 셸에 액세스하는 데 사용할 수 있습니다. **Live CD** 옵션은 설치하기 전에 FreeBSD를 사용해 볼 수 있습니다. 라이브 버전은 라이브 CD 사용하기에 설명되어 있습니다.

하드웨어 디바이스 프로브를 포함한 부팅 메시지를 검토하려면 대문자 또는 소문자 **S**를 누른 다음 **Enter** 키를 눌러 셸에 액세스합니다. 셸 프롬프트에서 more /var/run/dmesg.boot를 입력한 다음 스페이스바 를 사용하여 메시지를 스크롤합니다. 완료되면 exit를 입력하여 시작 메뉴로 돌아갑니다.



2.5. bsdinstall 사용하기

이 섹션에서는 bsdinstall 메뉴의 순서와 시스템을 설치하기 전에 묻는 정보 유형을 보여줍니다. 화살표 키를 사용하여 메뉴 옵션을 강조 표시한 다음 스페이스 키를 눌러 해당 메뉴 항목을 선택하거나 선택 취소합니다. 완료되면 **Enter** 키를 눌러 선택 내용을 저장하고 다음 화면으로 이동합니다.

2.5.1. 키맵 메뉴 선택

프로세스를 시작하기 전에 키맵 로딩에 표시된 대로 bsdinstall이 키맵 파일을 로드합니다.

Figure 4. Keymap Loading

키맵이 로드되면 bsdinstall은 키맵 선택 메뉴에 표시된 메뉴를 표시합니다. 위쪽 및 아래쪽 화살표를 사용하여 시스템에 연결된 키보드의 매핑을 가장 가깝게 나타내는 키맵을 선택합니다. **Enter** 키를 눌러 선택 사항을 저장합니다.

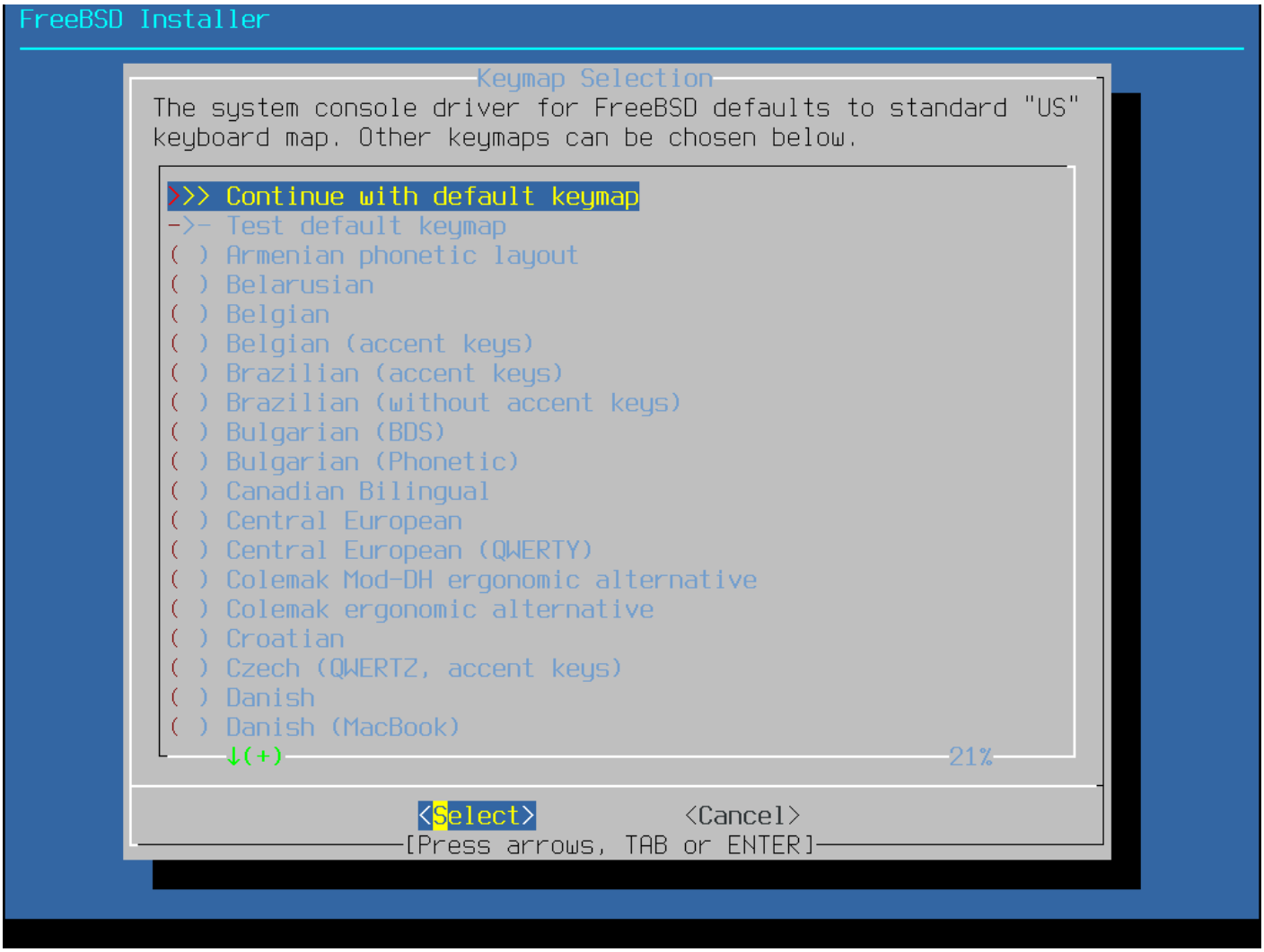


Figure 5. Keymap Selection Menu

Esc 키를 누르면 이 메뉴가 종료되고 기본 키맵이 사용됩니다 . 키맵 선택이 명확하지 않은 경우 미국 ISO-8859-1을 사용하는 것도 안전한 옵션입니다 .

또한 다른 키맵을 선택할 경우 키맵 테스트 메뉴에 표시된 것처럼 사용자가 키맵을 시도하고 올바른지 확인한 후 계속 진행할 수 있습니다 .

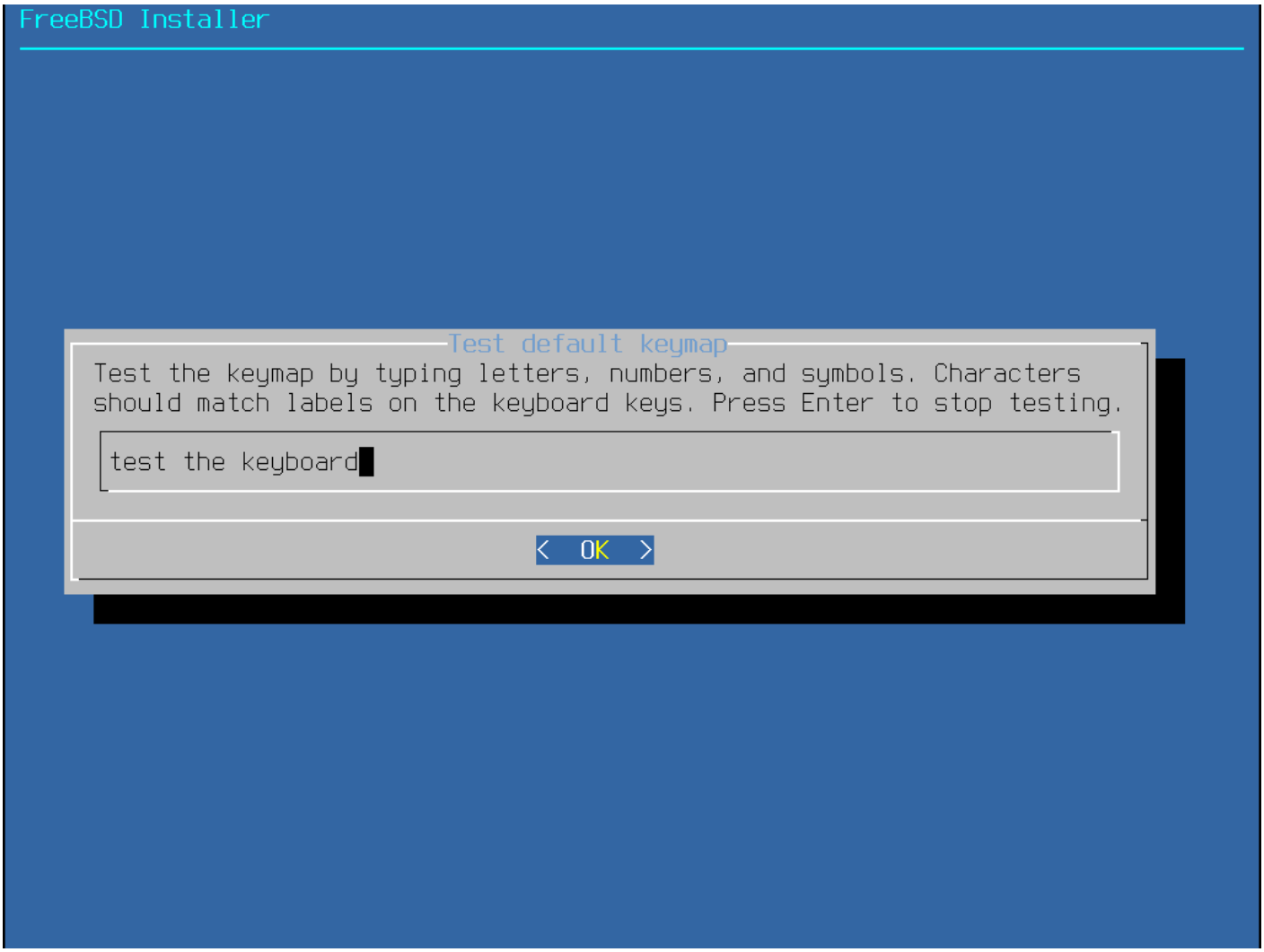


Figure 6. Keymap Testing Menu

2.5.2. 호스트 이름 설정

다음 bsdinstall 메뉴는 새로 설치한 시스템의 호스트 이름을 설정하는 데 사용됩니다 .

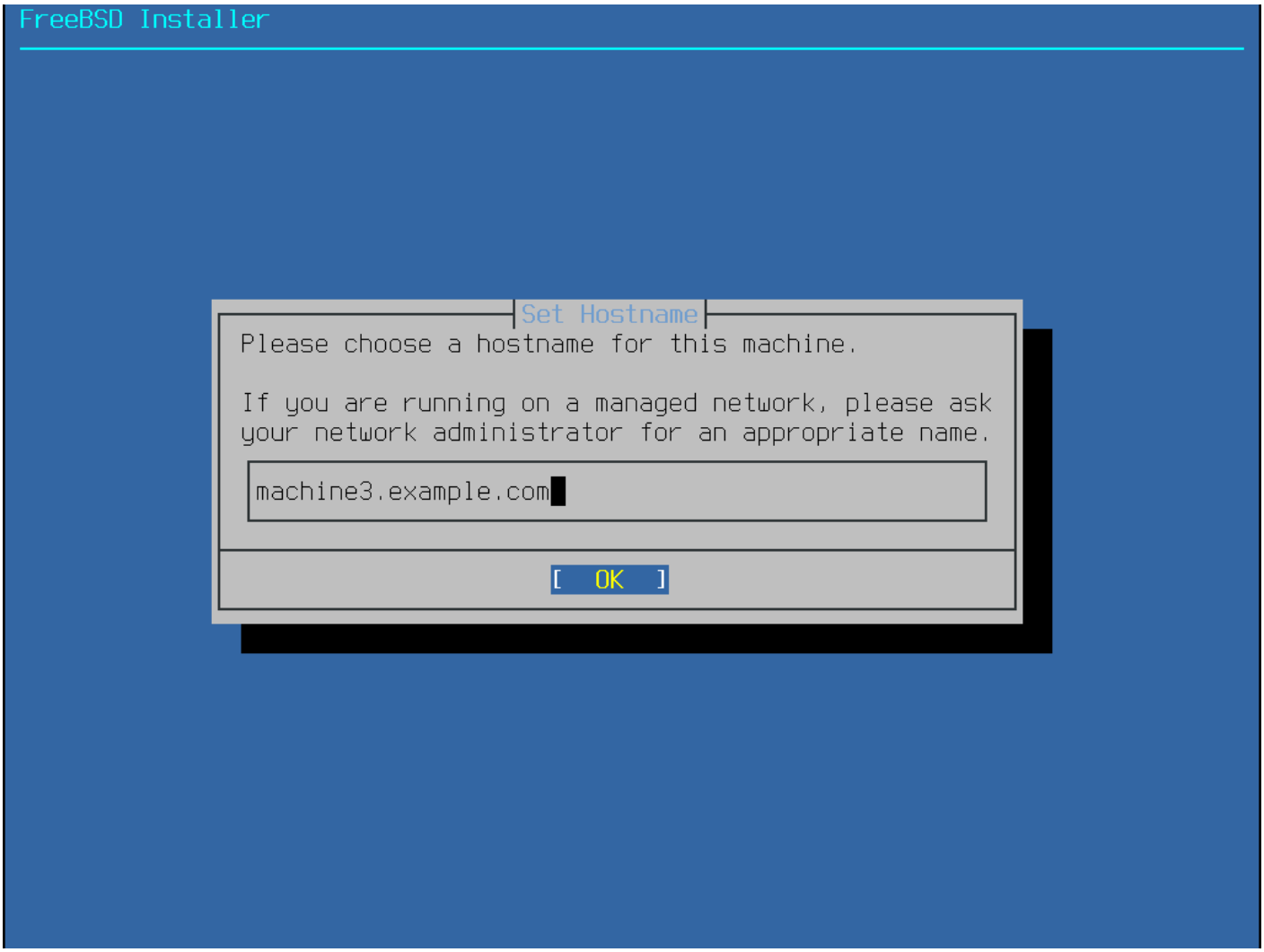


Figure 7. Setting the Hostname

네트워크에 고유한 호스트 이름을 입력합니다 . machine3.example.com과 같이 정규화된 호스트 이름이어야 합니다 .

2.5.3. 설치할 컴포넌트 선택

다음으로 bsdinstall은 설치할 선택적 구성 요소를 선택하라는 메시지를 표시합니다 .

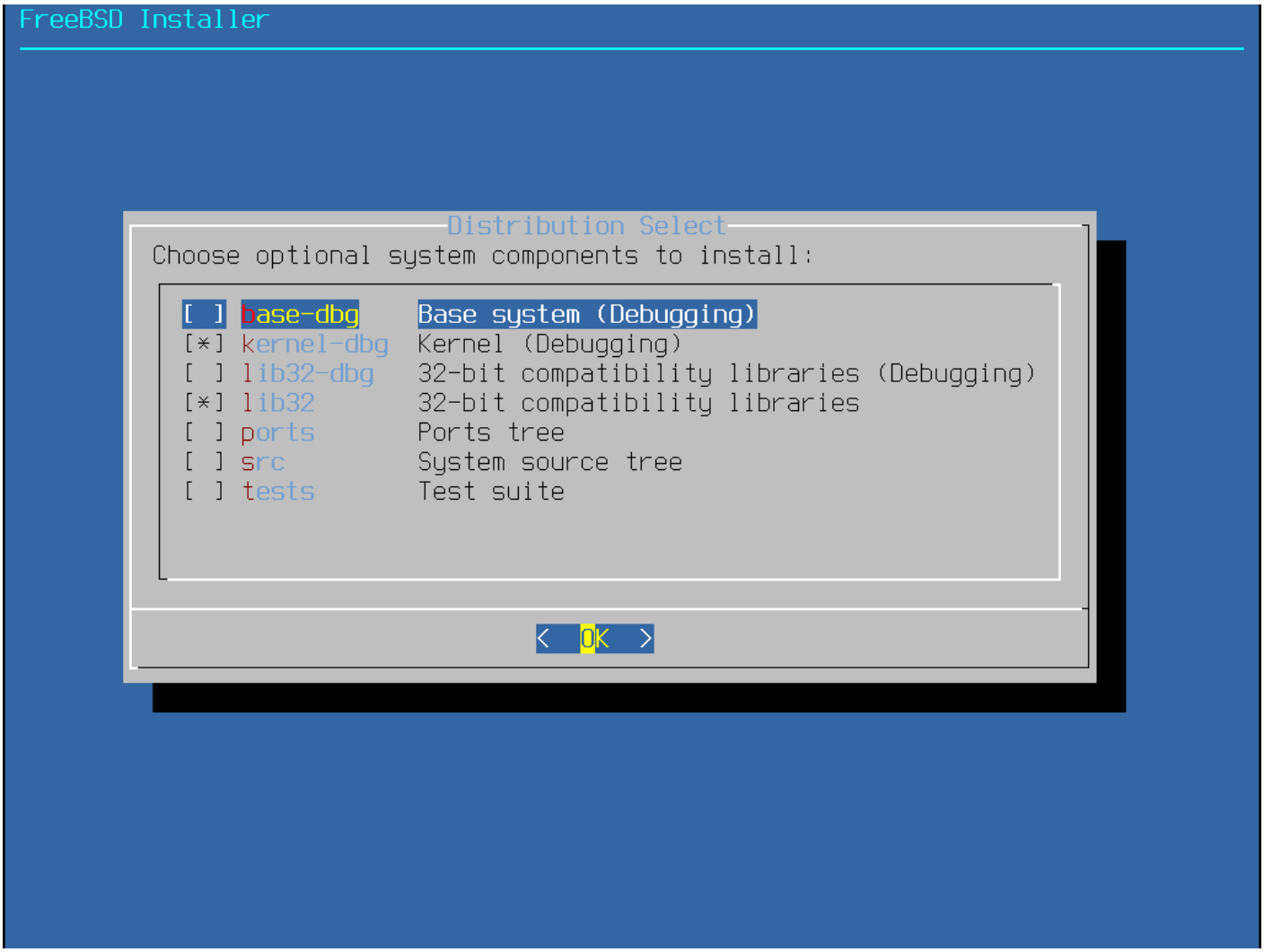


Figure 8. Selecting Components to Install

설치할 구성 요소를 결정하는 것은 시스템의 용도와 사용 가능한 디스크 공간의 양에 따라 크게 달라집니다 . *base system*으로 통칭되는 FreeBSD 커널과 유저랜드는 항상 설치됩니다 . 아키텍처에 따라 이러한 구성 요소 중 일부는 나타나지 않을 수 있습니다 :

- **base-dbg** - 디버그 심볼이 활성화된 cat 및 ls와 같은 기본 도구 .
- **kernel-dbg** - 디버그 심볼이 활성화된 커널 및 모듈 .
- **lib32-dbg** - 디버그 심볼이 활성화된 64비트 버전의 FreeBSD에서 32비트 애플리케이션을 실행하기 위한 호환성 라이브러리 .
- **lib32** - 64비트 버전의 FreeBSD에서 32비트 애플리케이션을 실행하기 위한 호환성 라이브러리 .
- **ports** - FreeBSD 포트 컬렉션은 타사 소프트웨어 패키지의 다운로드 , 컴파일 및 설치를 자동화하는 파일 모음입니다 . 애플리케이션 설치 : 패키지와 포트에서는 포트 컬렉션을 사용하는 방법에 대해 설명합니다 .

설치 프로그램에서 디스크 공간이 충분한지 확인하지 않습니다 . 충분한 하드 디스크 공간을 사용할 수 있는 경우에만 이 옵션을 선택하십시오 . FreeBSD 포트 컬렉션은 약 3GB의 디스크 공간을 차지합니다 .

- **src** - [illegible] FreeBSD [illegible] .
- **tests** - FreeBSD [illegible] .

2.5.4. [illegible]

[illegible] -bootonly.iso [illegible] -mini-memstick.img[illegible] .

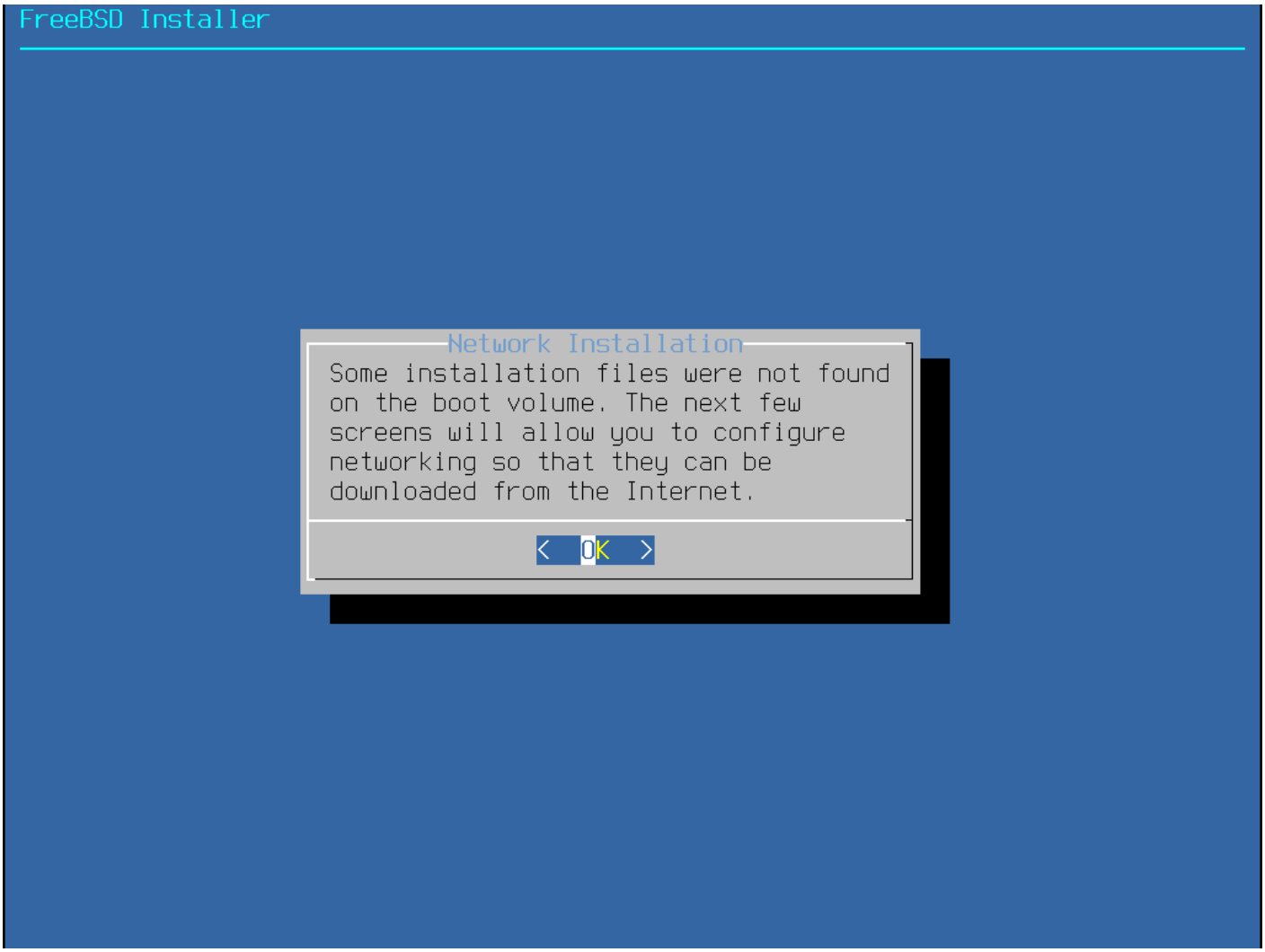
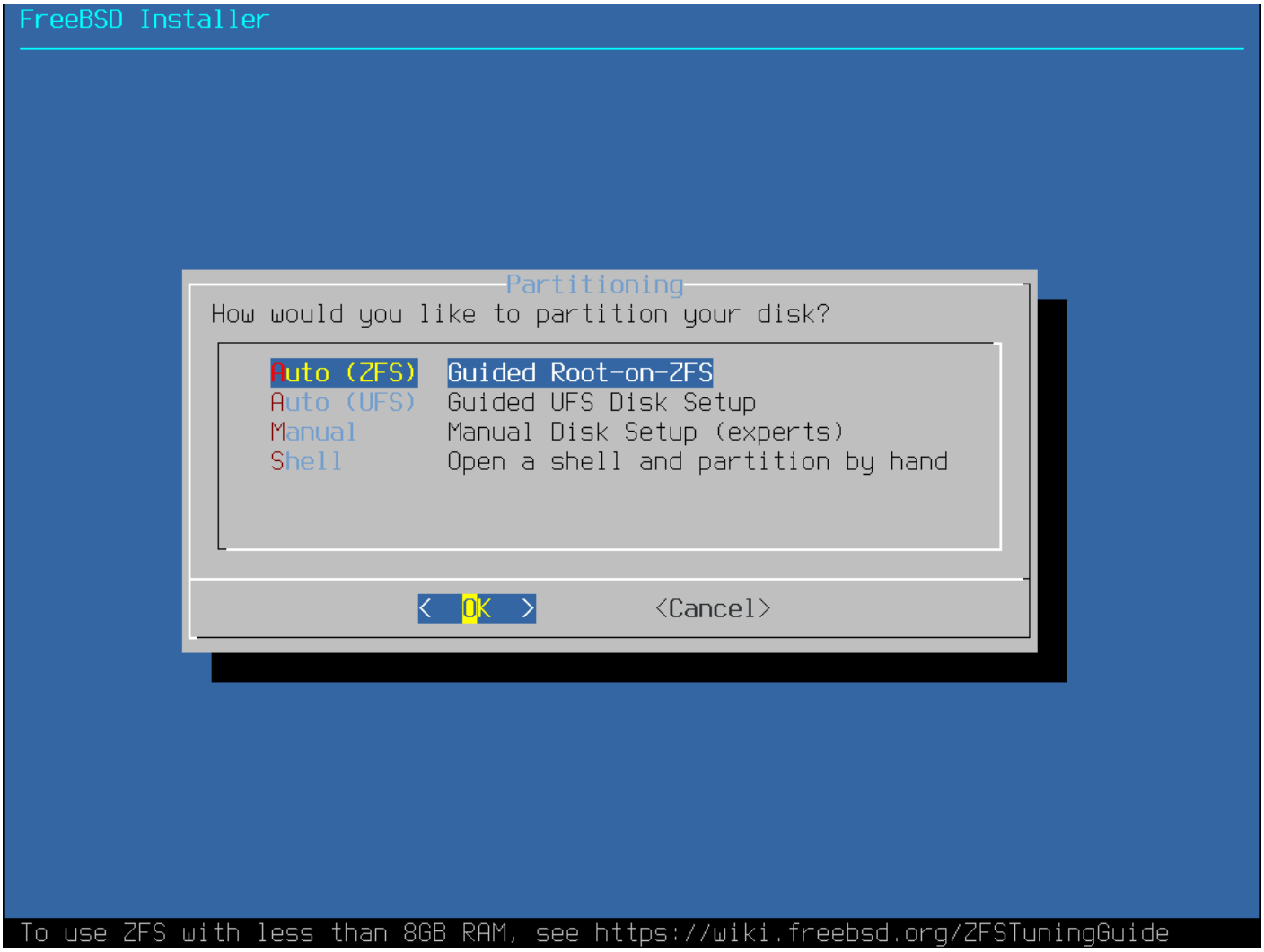


Figure 9. Installing from the Network



2.6. 디스크 공간 할당

다음 메뉴는 디스크 공간을 할당하는 방법을 결정하는 데 사용됩니다 .

Figure 10. Partitioning Choices

bsdinstall은 사용자에게 디스크 공간을 할당하는 네 가지 방법을 제공합니다 :

- **Auto (ZFS)** 파티셔닝은 부팅 환경을 위한 선택적 GELI 암호화를 지원하는 루트 온 ZFS 시스템을 생성합니다 .
- **Auto (UFS)** 파티셔닝은 UFS 파일 시스템을 사용하여 디스크 파티션을 자동으로 설정합니다 .
- **Manual** 파티셔닝을 사용하면 고급 사용자가 메뉴 옵션에서 사용자 지정 파티션을 만들 수 있습니다 .
- **Shell**에서는 고급 사용자가 gpart(8), fdisk(8), bsdlabel(8) 등의 명령줄 유틸리티를 사용하여 사용자 지정 파티션을 만들 수 있는 셸 프롬프트가 열립니다 .

이 섹션에서는 디스크 파티션을 배치할 때 고려해야 할 사항을 설명합니다 . 그런 다음 다양한 파티셔닝 방법을 사용하는 방법을 설명합니다 .

2.6.1. 파티션 레이아웃 디자인하기

파일 시스템의 기본 파티션 레이아웃은 전체 시스템에 대해 하나의 파일 시스템을 포함합니다 . UFS를 사용할 때 디스크 공간이 충분하거나 디스크가 여러 개인 경우 여러 파일 시스템 사용을 고려할 수 있습니다 . 파일 시스템을 배치할 때는 하드 드라이브가 바깥쪽 트랙에서 안쪽 트랙으로 데이터를 더 빨리 전송한다는 점을 기억하세요 . 따라서 크기가 작고 액세스 빈도가 높은 파일 시스템은 드라이브 바깥쪽에 가깝게 배치하고 /usr 과 같은 큰 파티션은 디스크 안쪽을 향해 배치해야 합니다 . 다음과 비슷한 순서로 파티션을 만드는 것이 좋습니다 : /, swap, /var, /usr과 비슷한 순서로 파티션을 만드는 것이 좋습니다 .

/var 파티션의 크기는 사용하려는 컴퓨터의 용도를 반영합니다 . 이 파티션은 사서함 , 로그 파일 및 프린터 스풀을 보관하는 데 사용됩니다 . 사서함과 로그 파일은 사용자 수와 로그 파일의 보관 기간에 따라 예기치 않은 크기로 커질 수 있습니다 . 평균적으로 대부분의 사용자는 /var에 1기가바이트 이상의 디스크 여유 공간이 필요한 경우는 거의 없습니다 .

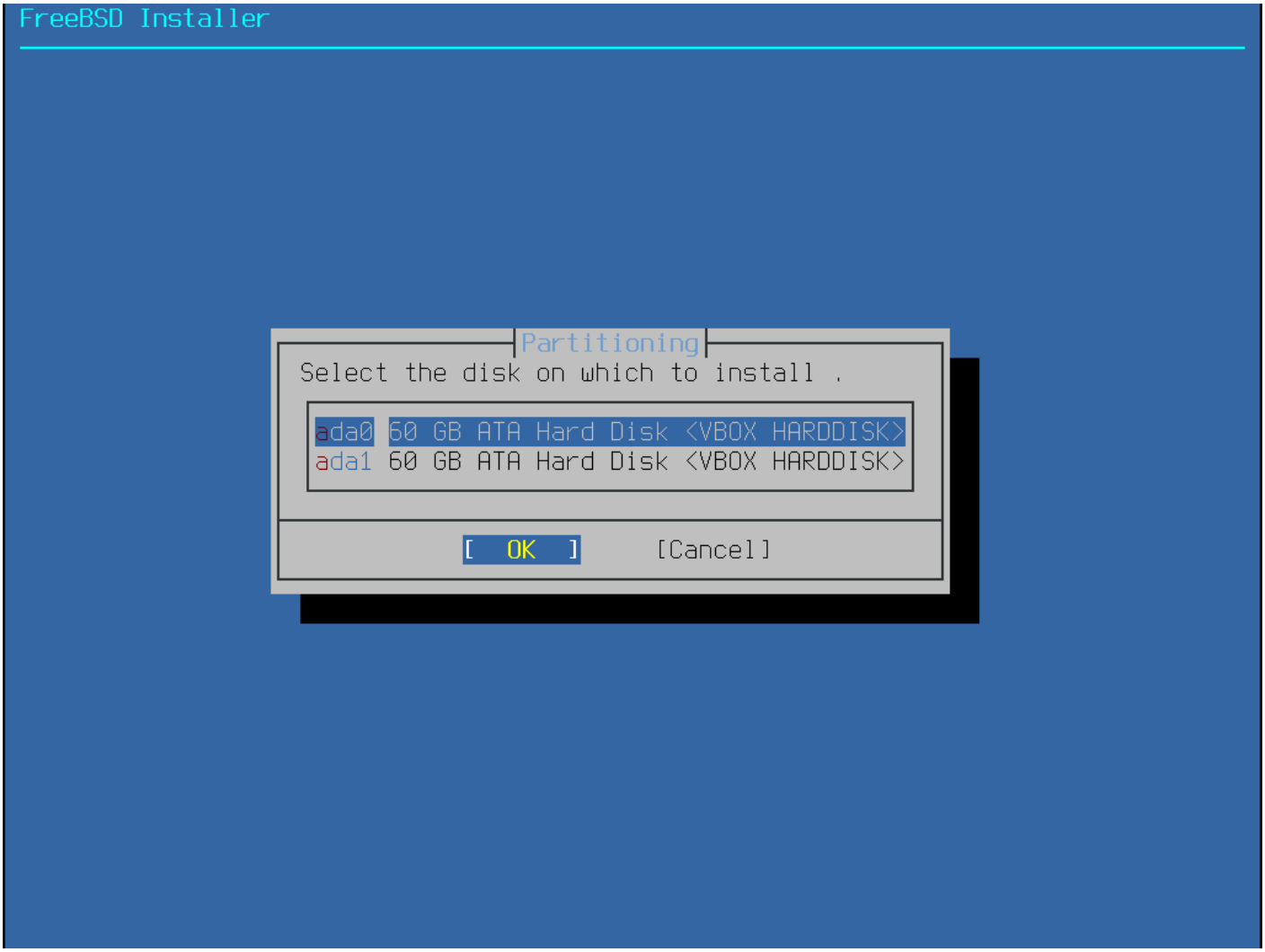
> 가끔 /var/tmp에 많은 디스크 공간이 필요할 때가 있습니다 . 새 소프트웨어가 설치되면 패키징 도구는 /var/tmp에서 패키지의 임시 복사본을 추출합니다 . Firefox나 LibreOffice 와 같은 대용량 소프트웨어 패키지는 /var/tmp에 디스크 공간이 충분하지 않으면 설치가 까다로울 수 있습니다 .

/usr 파티션에는 FreeBSD 포트 컬렉션과 시스템 소스 코드를 포함하여 시스템을 지원하는 많은 파일이 저장됩니다 . 이 파티션에는 최소 2기가바이트의 공간이 권장됩니다 . 또한 사용자의 홈 디렉토리는 기본적으로 /usr/home에 배치되지만 다른 파티션에 배치할 수 있습니다 . 기본적으로 /home은 /usr/home에 대한 심볼릭 링크입니다 .

파티션 크기를 선택할 때는 공간 요구 사항을 염두에 두세요 . 한 파티션은 거의 사용하지 않는데 다른 파티션의 공간이 부족하면 번거로울 수 있습니다 .

일반적으로 스왑 파티션은 실제 메모리 (RAM) 크기의 약 2배가 되어야 합니다 . RAM이 최소인 시스템 (대용량 메모리 구성의 경우 더 적은 메모리)은 스왑을 더 많이 설정하면 성능이 더 좋아질 수 있습니다 . 스왑을 너무 적게 구성하면 VM 페이지 스캔 코드의 비효율성을 초래할 수 있으며 나중에 메모리를 더 추가할 경우 문제가 발생할 수 있습니다 .

여러 개의 SCSI 디스크 또는 여러 개의 IDE 디스크가 서로 다른 컨트롤러에서 작동하는 대규모 시스템에서는 각 드라이브에 최대 4개의 드라이브까지 스왑을 구성하는 것이 좋습니다 . 스왑 파티션은 거의 같은 크기여야 합니다 . 커널은 임의의 크기를 처리할 수 있지만 내부 데이터 구조는 가장 큰 스왑 파티션의 4배까지 확장됩니다 . 스왑 파티션을 거의 같은 크기로 유지하면 커널이 디스크에서 스왑 공간을 최적으로 스트라이핑할 수 있습니다 . 스왑 크기가 크면 구성된 총 스왑에 대한 커널 경고 메시지가 표시될 수 있습니다 . 경고 메시지의 지시에 따라 스왑 할당을 추적하는 데 허용되는 메모리 양을 늘리면 제한이 높아집니다 . 강제로



재부팅하기 전에 런어웨이 프로그램에서 복구하는 것이 더 쉬울 수 있습니다 .

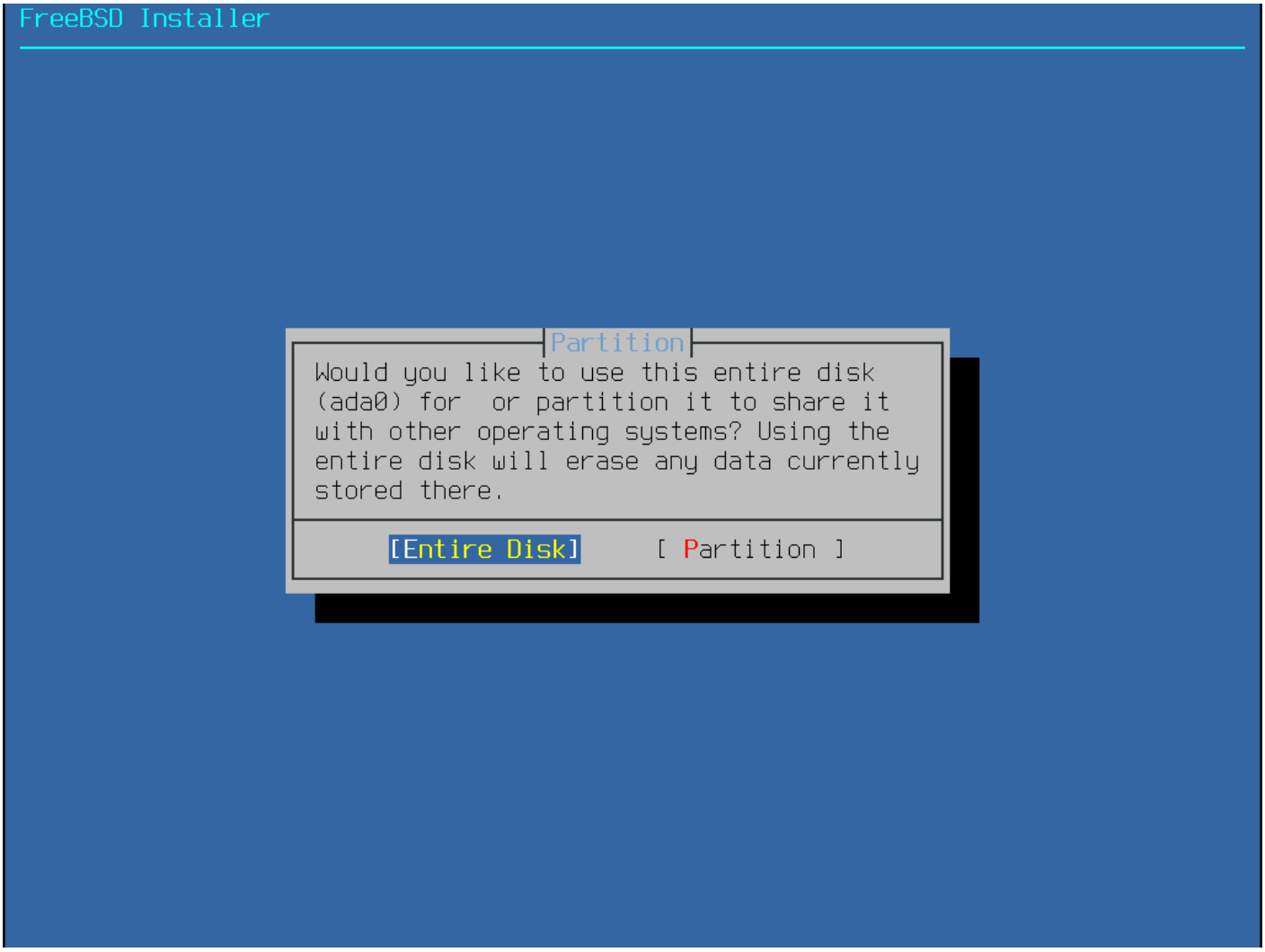
시스템을 적절하게 파티셔닝하면 쓰기가 많은 작은 파티션에서 발생한 조각화가 대부분 읽기가 많은 파티션으로 번지지 않습니다 . 쓰기가 많은 파티션을 디스크 가장자리에 더 가깝게 유지하면 쓰기 작업이 가장 많이 발생하는 파티션의 I/O 성능이 향상됩니다 . 더 큰 파티션의 I/O 성능이 필요할 수 있지만 , 파티션을 디스크 가장자리로 더 이동한다고 해서 /var를 가장자리로 이동하는 것보다 성능이 크게 향상되지는 않습니다 .

2.6.2. UFS를 사용한 가이드 파티셔닝

이 방법을 선택하면 메뉴에 사용 가능한 디스크가 표시됩니다 . 여러 개의 디스크가 연결되어 있는 경우 FreeBSD를 설치할 디스크를 선택합니다 .

Figure 11. Selecting from Multiple Disks

디스크를 선택하면 다음 메뉴에서 **Entire Disk**에 설치하거나 여유 공간을 사용하여 파티션을 만들지 묻는 메시지가 표시됩니다 . 전체 디스크를 선택하면 디스크 전체를 채우는 일반 파티션 레이아웃이 자동으로



만들어집니다 . **Partition**을 선택하면 디스크의 사용되지 않는 공간에서 파티션 레이아웃이 만들어집니다 .

Figure 12. Selecting Entire Disk or Partition

Entire Disk 옵션을 선택하면 디스크가 지워질 것임을 나타내는 대화 상자가 bsdinstall에 표시됩니다 .

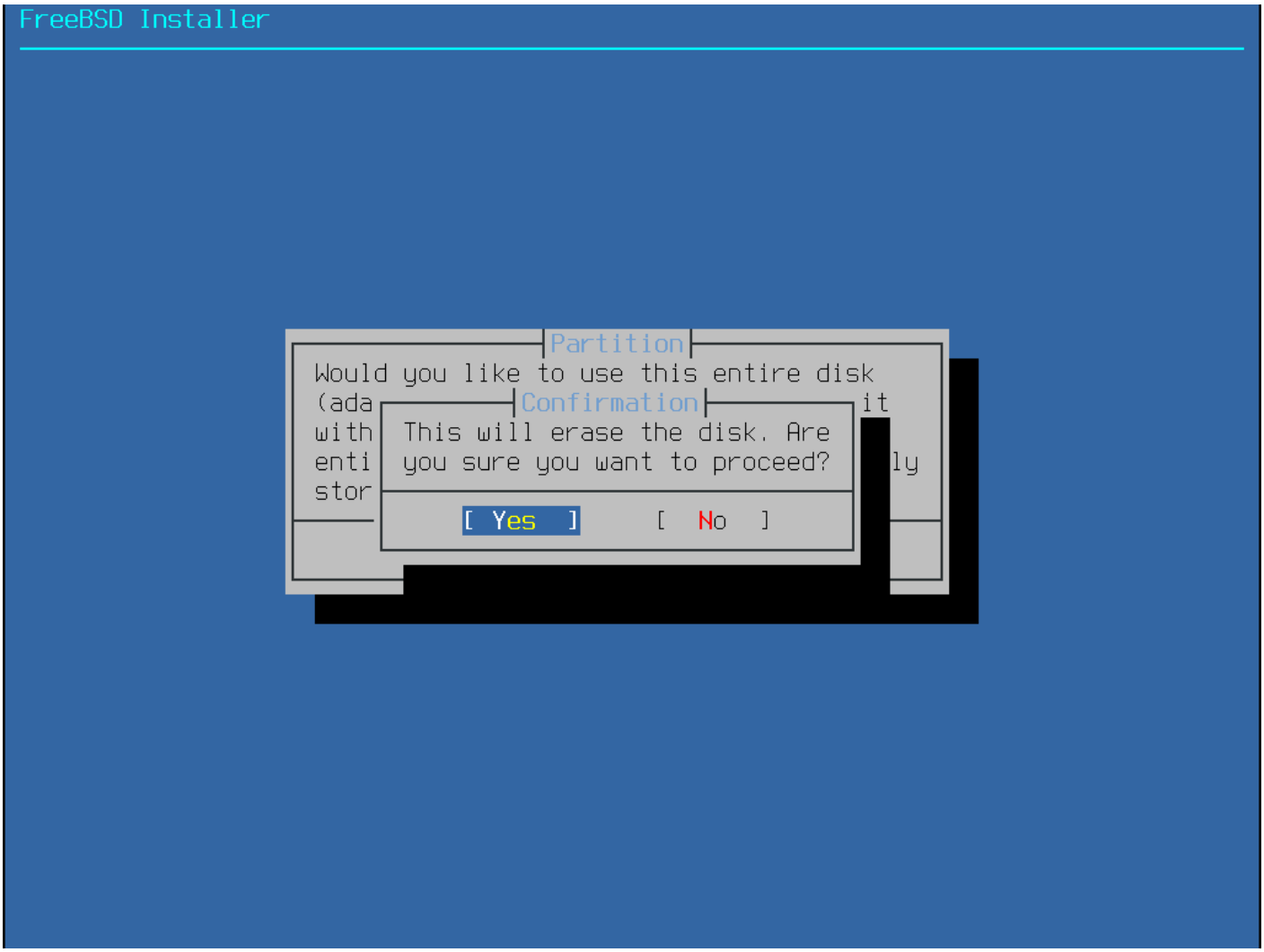


Figure 13. Confirmation

다음 메뉴에는 사용 가능한 파티션 구성표 유형 목록이 표시됩니다 . 일반적으로 amd64 컴퓨터에는 GPT가 가장 적합한 선택입니다 . GPT와 호환되지 않는 구형 컴퓨터는 MBR을 사용해야 합니다 . 다른 파티션 구성표는 일반적으로 일반적이지 않거나 구형 컴퓨터에 사용됩니다 . 자세한 내용은 파티션 구성표에서 확인할 수 있습니다 .

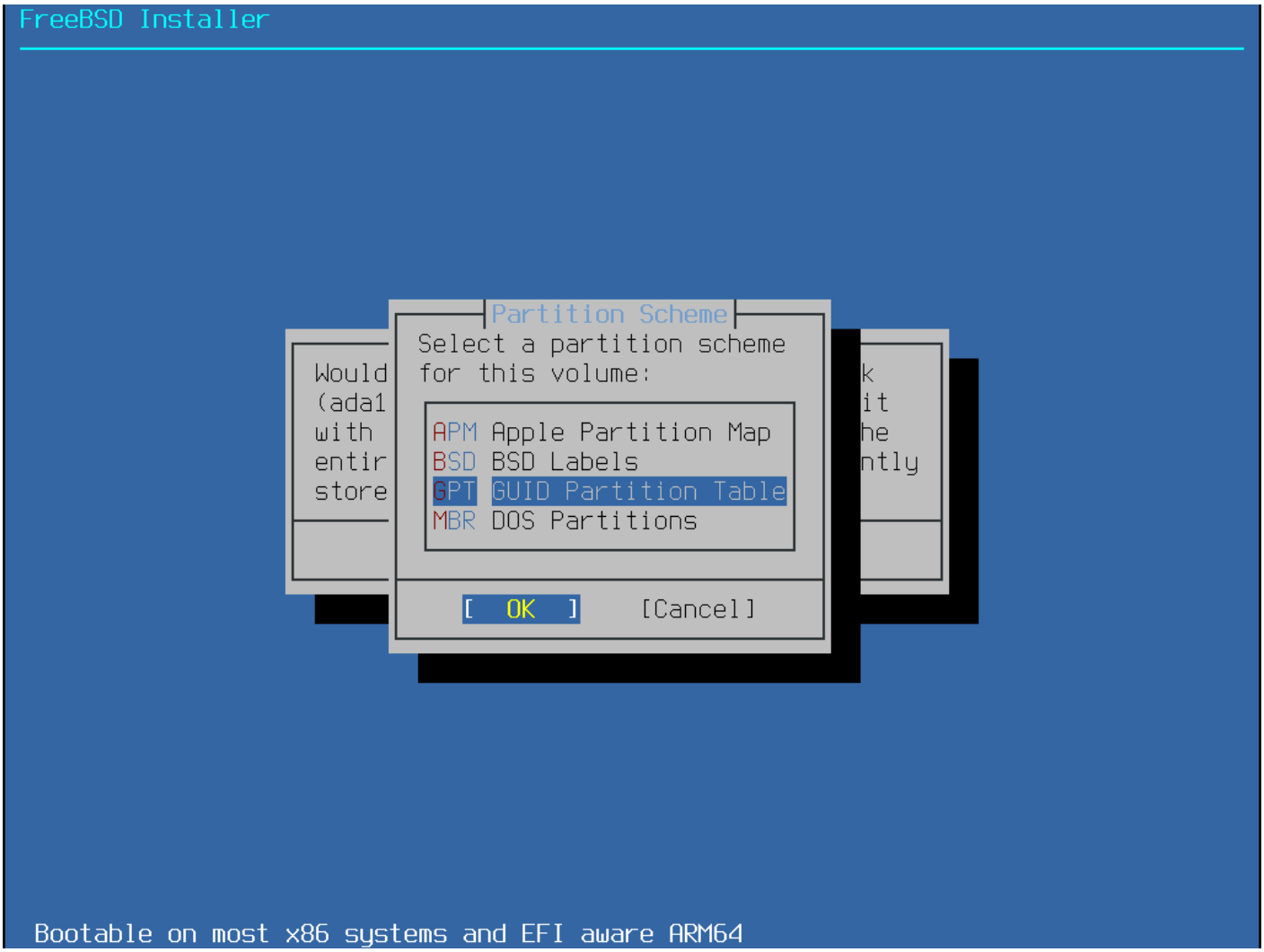


Figure 14. Select Partition Scheme

파티션 레이아웃이 생성된 후에는 이를 검토하여 설치 요구 사항을 충족하는지 확인합니다 . 되돌리기를 선택하면 파티션이 원래 값으로 재설정됩니다 . 자동을 누르면 자동으로 FreeBSD 파티션이 다시 생성됩니다 . 파티션을 수동으로 생성 , 수정 또는 삭제할 수도 있습니다 . 파티션 분할이 올바르면 마침을 선택하여 설치를 계속 진행합니다 .

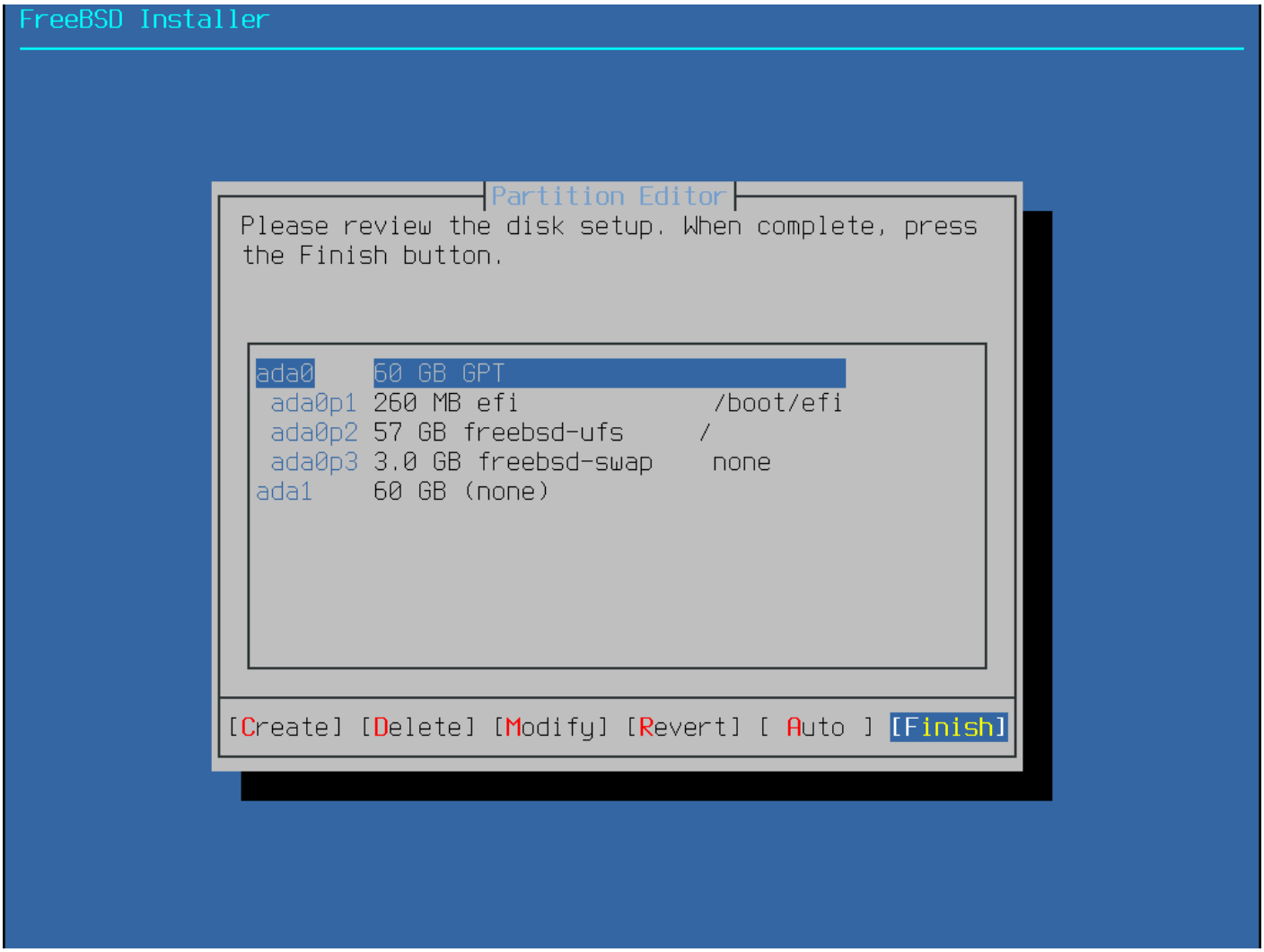


Figure 15. Review Created Partitions

디스크 구성이 완료되면 다음 메뉴에서 선택한 드라이브를 포맷하기 전에 마지막으로 변경할 수 있는 기회를 제공합니다 . 변경이 필요한 경우 , **Back**를 선택하여 기본 파티셔닝 메뉴로 돌아갑니다 . **Revert & Exit**는 드라이브를 변경하지 않고 설치 관리자를 종료합니다 . 그렇지 않으면 **Commit**을 선택하여 설치 프로세스를 시작합니다 .

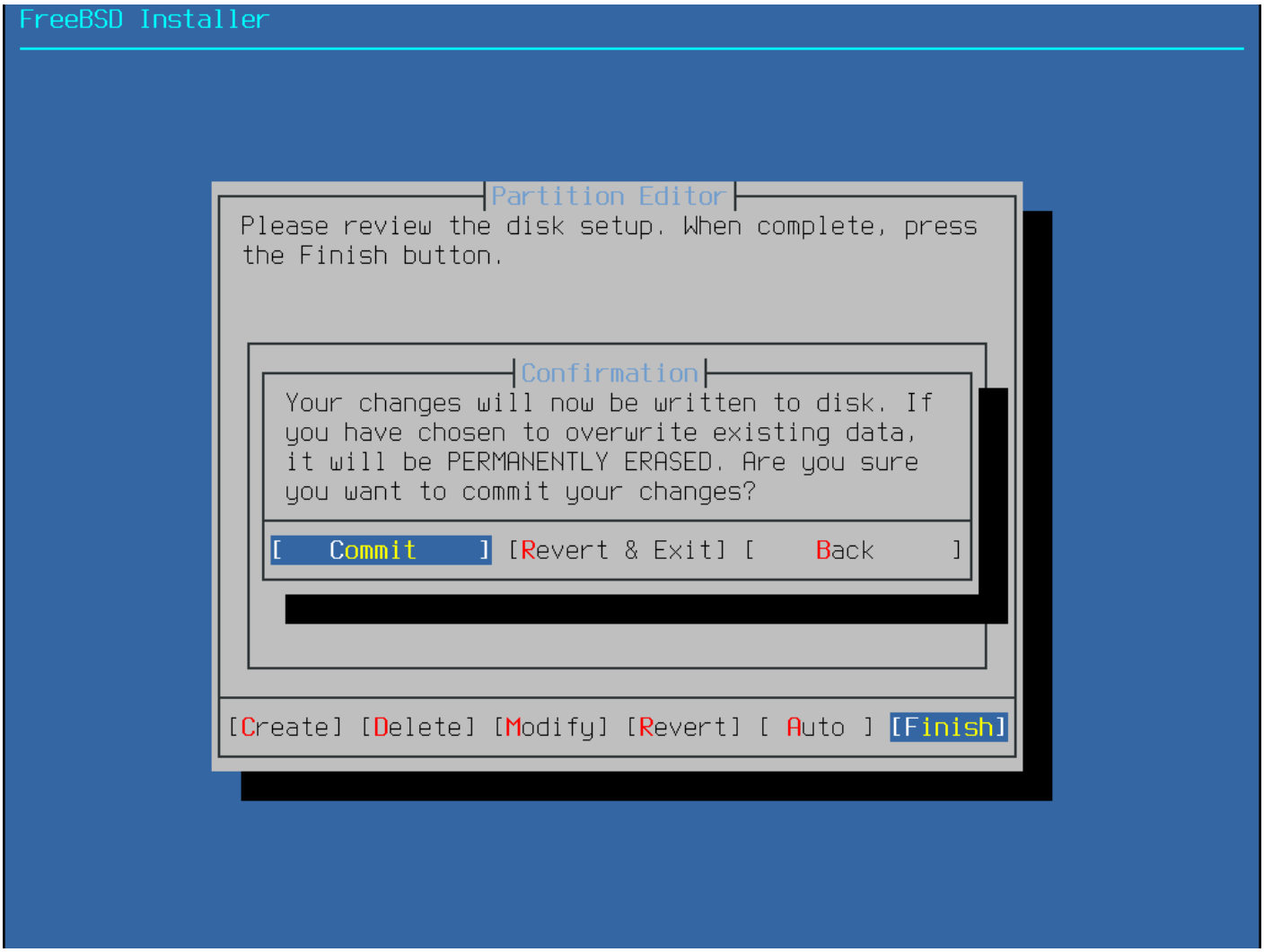


Figure 16. Final Confirmation

설치 프로세스를 계속 진행하려면 배포 파일 가져오기로 이동합니다 .

2.6.3. 수동 파티셔닝

이 방법을 선택하면 파티션 편집기가 열립니다 :

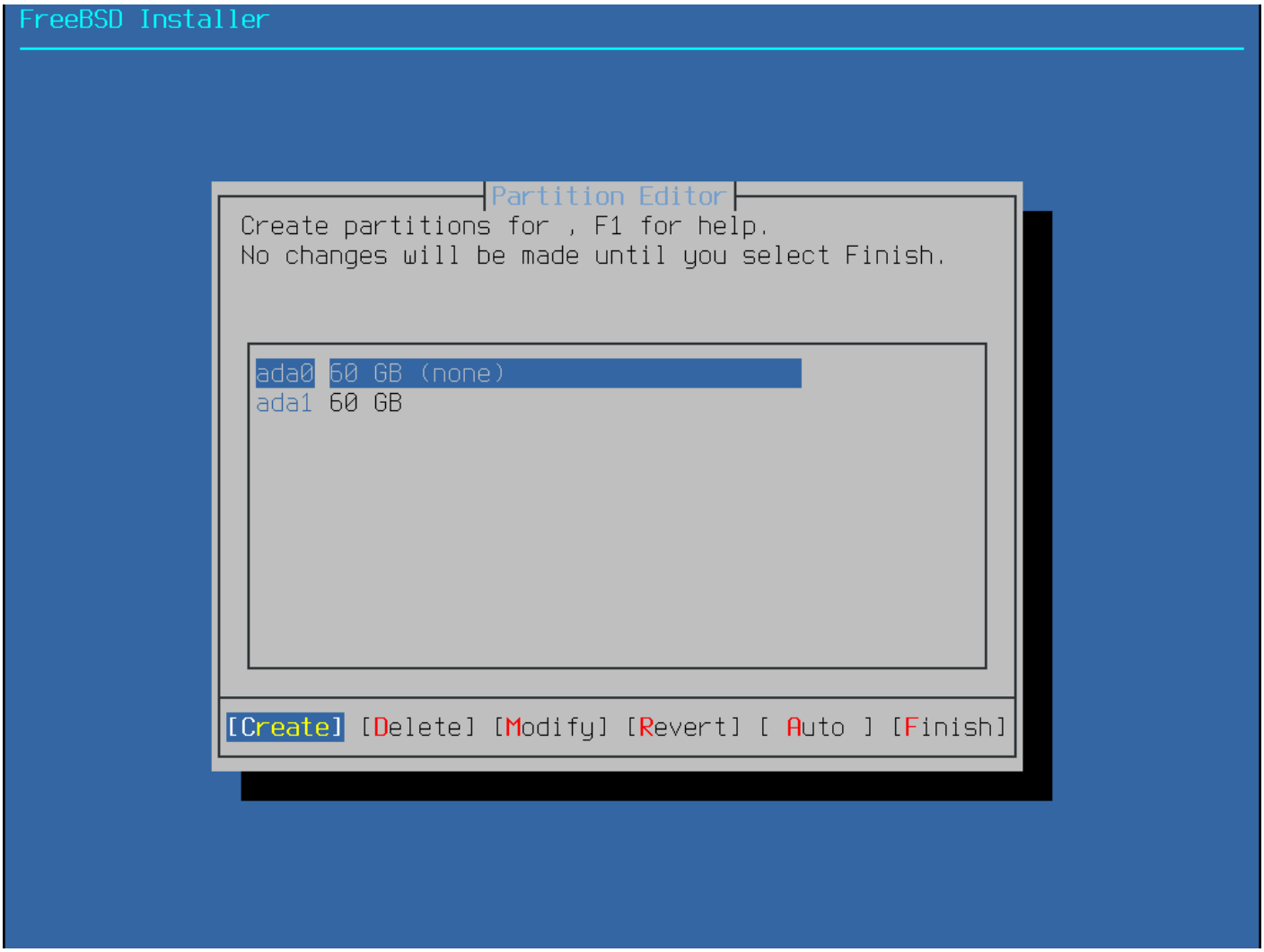


Figure 17. Manually Create Partitions

설치 드라이브 (이 예에서는 ada0)를 강조 표시하고 **Create**를 선택하여 사용 가능한 파티션 구성표 메뉴를 표시합니다 :

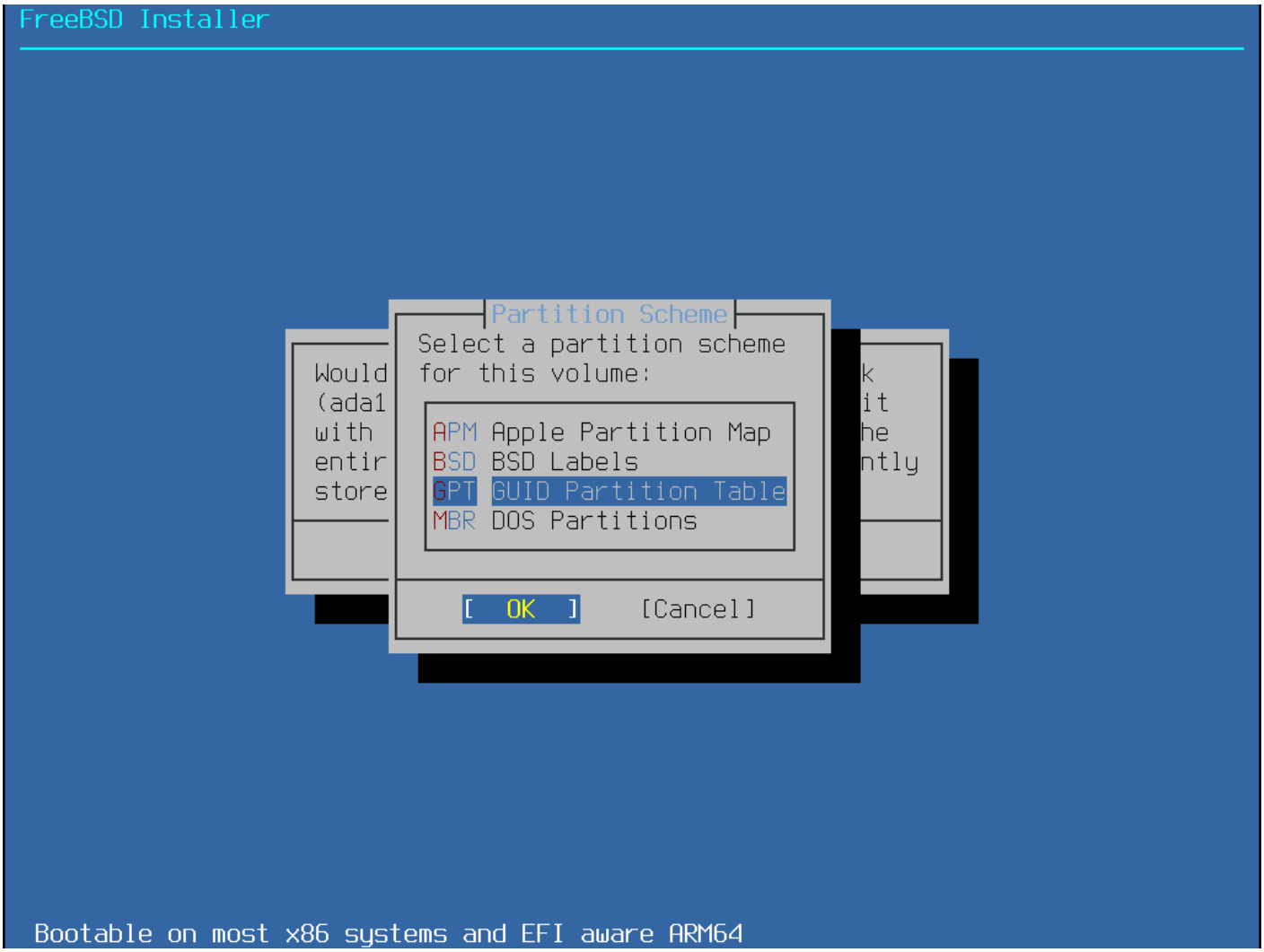


Figure 18. Manually Create Partitions

일반적으로 amd64 컴퓨터에는 GPT가 가장 적합한 선택입니다. GPT와 호환되지 않는 구형 컴퓨터는 MBR을 사용해야 합니다. 다른 파티션 체계는 일반적으로 일반적이지 않거나 구형 컴퓨터에 사용됩니다.

Table 1. Partitioning Schemes

Abbreviation	Description
APM	PowerPC®에서 사용하는 Apple 파티션 맵.
BSD	non-BSD 디스크 유틸리티가 인식하지 못할 수 있으므로 위험 전용 모드(dangerously dedicated mode)라고도 하는 MBR이 없는 BSD 레이블입니다.
GPT	GUID Partition Tables
MBR	Master Boot Record.

파티션 구성표를 선택하고 생성한 후 다시 **Create**를 선택하여 파티션을 생성합니다. **Tab** 키를 사용하여 **<OK>**, **<Options>**, **<Cancel>**를 순환한 후 필드에 초점을 맞출 수 있습니다 (**<OK>**, **<Options>**, **<Cancel>**를 순환한 후).

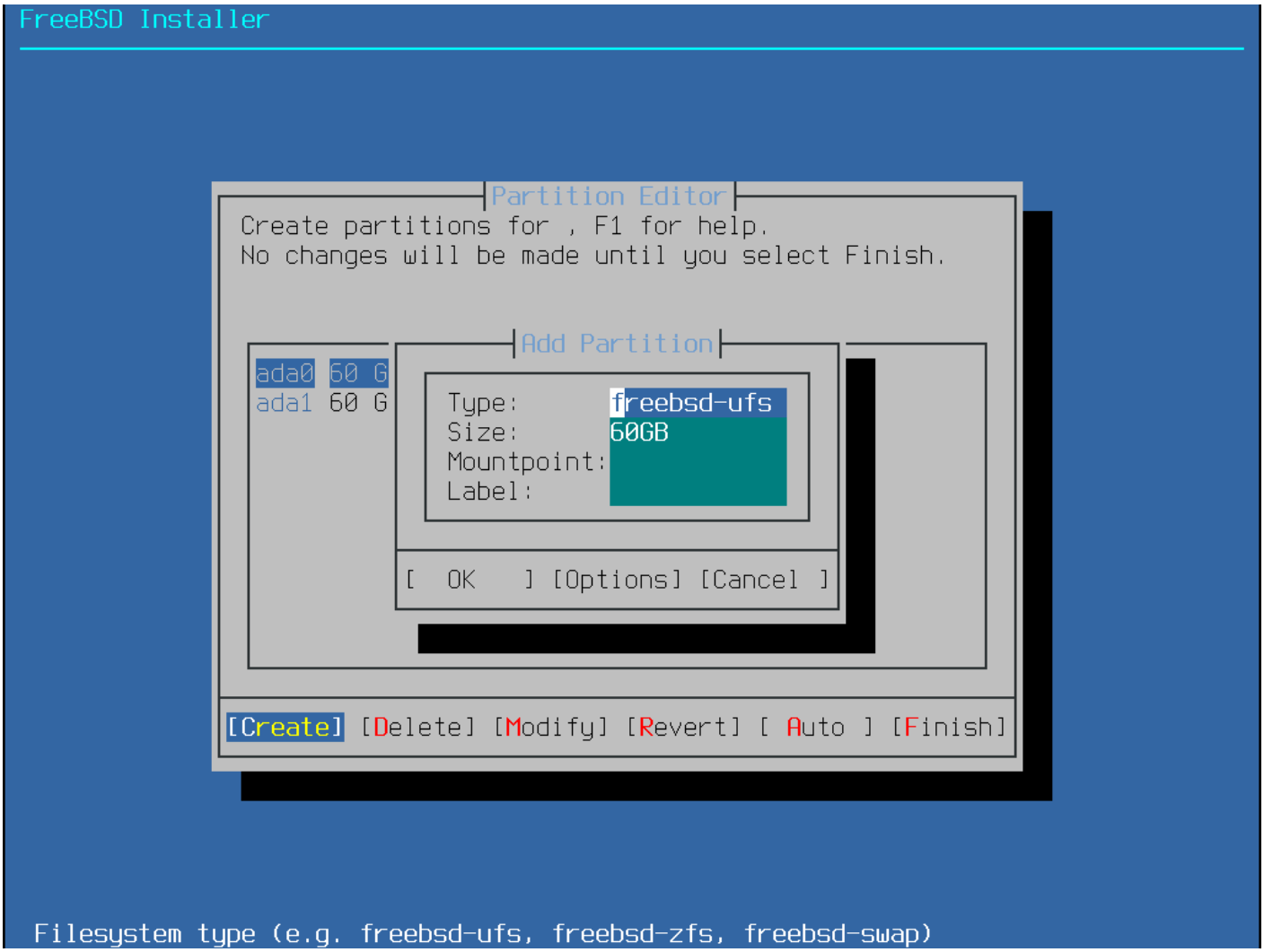


Figure 19. Manually Create Partitions

표준 FreeBSD GPT 설치는 UFS 또는 ZFS를 포함하여 최소 3개의 파티션을 사용합니다 :

- freebsd-boot 또는 efi - FreeBSD 부팅 코드가 들어 있습니다 .
- freebsd-ufs - FreeBSD UFS 파일 시스템 .
- freebsd-zfs - FreeBSD ZFS 파일 시스템 . ZFS에 대한 자세한 정보는 Z 파일 시스템 (ZFS)에서 확인할 수 있습니다 .
- freebsd-swap - FreeBSD 스왑 공간 .

사용 가능한 GPT 파티션 유형에 대한 설명은 gpart(8)을 참조하세요 .

여러 개의 파일 시스템 파티션을 만들 수 있습니다 . 어떤 사람들은 /, /var, /tmp 및 /usr에 대해 별도의 파티션이 있는 기존 레이아웃을 선호합니다 .

메모리가 충분한 시스템에서는 나중에 메모리 기반 파일 시스템 (tmpfs(5))으로 /tmp를 추가할 수 있다는 점에 유의하세요 .

예는 기존 분할 파일 시스템 파티션 만들기를 참조하세요 .

Size는 일반적인 약어로 입력할 수 있습니다 : K는 킬로바이트 , M은 메가바이트 , G는 기가바이트를 입력할 수 있습니다 .

TIP

적절한 섹터 정렬은 최상의 성능을 제공하며 , 파티션 크기를 4K 바이트의 배수로 만들면 512바이트 또는 4K 바이트 섹터가 있는 드라이브에서 정렬을 보장하는 데 도움이 됩니다 . 일반적으로 1M 또는 1G의 짝수 배수인 파티션 크기를 사용하는 것이 모든 파티션이 4K의 짝수 배수에서 시작하도록 하는 가장 쉬운 방법입니다 . 한 가지 예외가 있는데 , 현재 부팅 코드 제한으로 인해 freebsd-boot 파티션은 512K보다 크지 않아야 합니다 .

파티션에 파일 시스템이 포함될 경우 Mountpoint가 필요합니다 . 하나의 UFS 파티션만 생성할 경우 마운트 지점은 /이어야 합니다 .

Label은 파티션을 알릴 수 있는 이름입니다 . 드라이브가 다른 컨트롤러나 포트에 연결된 경우 드라이브 이름이나 번호는 변경될 수 있지만 파티션 레이블은 변경되지 않습니다 . /etc/fstab과 같은 파일에서 드라이브 이름과 파티션 번호 대신 레이블을 참조하면 시스템이 하드웨어 변경에 더 잘 견딜 수 있습니다 . GPT 레이블은 디스크가 연결되면 /dev/gpt/에 표시됩니다 . 다른 파티셔닝 방식은 레이블 기능이 다르며 레이블은 /dev/의 다른 디렉터리에 표시됩니다 .

TIP

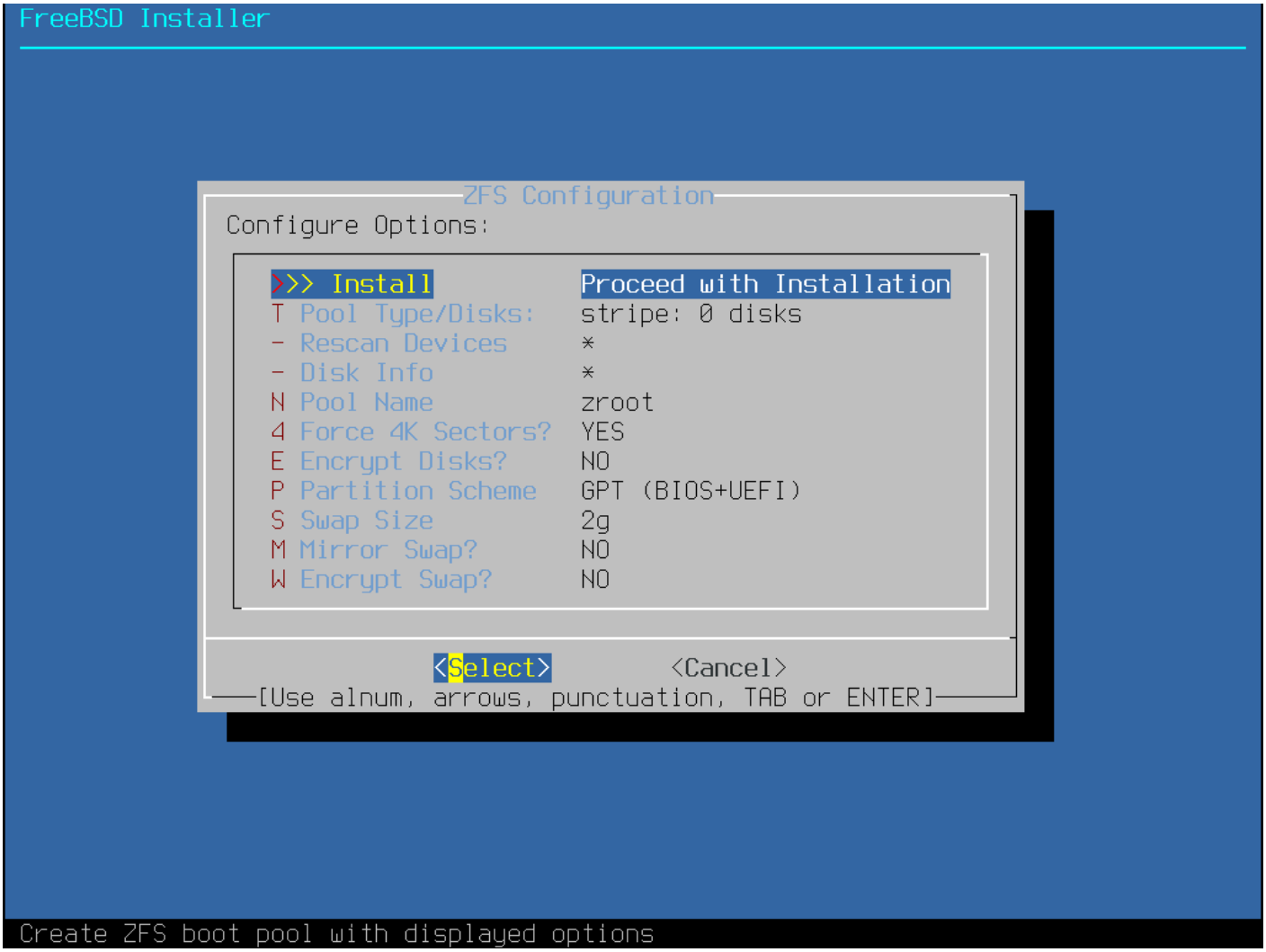
동일한 레이블로 인한 충돌을 피하려면 모든 파티션에 고유한 레이블을 사용하세요 . 컴퓨터의 이름 , 용도 또는 위치에서 몇 글자를 라벨에 추가할 수 있습니다 . 예를 들어 , 이름이 **lab**인 컴퓨터의 UFS 루트 파티션에는 **labroot** 또는 **rootfslab**을 사용합니다 .

예제 **1.** 기존 분할 파일 시스템 파티션 만들기

디렉터리 **/**, **/var**, **/tmp** 및 **/usr**이 자체 파티션에 별도의 파일 시스템인 기존 파티션 레이아웃의 경우 , GPT 파티셔닝 체계를 만든 다음 그림과 같이 파티션을 만듭니다 . 표시된 파티션 크기는 20G 대상 디스크의 일반적인 크기입니다 . 대상 디스크에 더 많은 공간을 사용할 수 있는 경우 더 큰 스왑 또는 **/var** 파티션이 유용할 수 있습니다 . 여기에 표시된 레이블 앞에는 "example"을 나타내는 **ex**가 붙지만 , 위에서 설명한 대로 다른 고유 레이블 값을 사용해야 합니다 .

기본적으로 FreeBSD의 **gptboot**는 첫 번째 UFS 파티션이 **/** 파티션이 될 것으로 예상합니다 .

Partition Type	Size	Mountpoint	Label
freebsd-boot	512K		
freebsd-ufs	2G	/	exrootfs
freebsd-swap	4G		exswap
freebsd-ufs	2G	/var	exvarfs
freebsd-ufs	1G	/tmp	extmpfs



freebsd-ufs	기본값을 수락합니다 (디스크의 나머지 부분).	/usr	exusrfs

사용자 지정 파티션이 생성되면 **Finish**를 선택하여 설치를 계속하고 배포 파일 가져오기로 이동합니다 .

2.6.4. Root-on-ZFS를 사용한 가이드

파티셔닝

이 파티셔닝 모드는 전체 디스크에서만 작동하며 전체 디스크의 콘텐츠를 지웁니다 . 기본 ZFS 구성 메뉴에는 풀 생성을 제어할 수 있는 여러 가지 옵션이 있습니다 .

Figure 20. ZFS Partitioning Menu

다음은 이 메뉴의 옵션에 대한 요약입니다 :

- **Install** - 선택한 옵션으로 설치를 진행합니다 .

- **Pool Type/Disks** - Pool Type과 풀을 구성할 디스크를 구성합니다 . 자동 ZFS 설치 관리자는 현재 스트라이프 모드를 제외하고 단일 최상위 레벨 vdev 생성만 지원합니다 . 더 복잡한 풀을 만들려면 셸 모드 파티셔닝의 지침에 따라 풀을 만드세요 .

- **Rescan Devices** - 사용 가능한 디스크 목록을 다시 채웁니다 .

- **Disk Info** - 이 메뉴에서는 각 디스크의 파티션 테이블과 장치 모델 번호 및 일련 번호와 같은 다양한 기타 정보 (사용 가능한 경우)를 포함하여 각 디스크를 검사하는 데 사용할 수 있습니다 .

- **Pool Name** - 풀의 이름을 설정합니다 . 기본 이름은 zroot입니다 .

- **Force 4K Sectors?** - 4K 섹터를 강제로 사용합니다 . 기본적으로 설치 관리자는 4K 경계에 맞춰 파티션을 자동으로 생성하고 ZFS가 4K 섹터를 사용하도록 강제합니다 . 이 방법은 512바이트 섹터 디스크에서도 안전하며 , 512바이트 디스크에 생성된 풀에 향후에 추가 저장 공간 또는 고장난 디스크의 대체용으로 4K 섹터 디스크를 추가할 수 있다는 추가적인 이점이 있습니다 . 활성화 여부를 선택하려면 Enter 키를 누르세요 .

- **Encrypt Disks?** - 디스크 암호화는 사용자가 GELI를 사용하여 디스크를 암호화할 수 있도록 합니다 . 디스크 암호화에 대한 자세한 내용은 "GELI를 사용한 디스크 암호화 "에서 확인할 수 있습니다 . Enter 키를 눌러 활성화할지 여부를 선택합니다 .

- **Partition Scheme** - 파티션 구성표를 선택합니다 . 대부분의 경우 GPT가 권장 옵션입니다 . 다른 옵션 중에서 선택하려면 Enter 키를 누릅니다 .
- **Swap Size** - 스왑 공간의 양을 설정합니다 .

- **Mirror Swap?** - 디스크 간 스왑을 미러링할지 여부입니다 . 미러 스왑을 활성화하면 크래시 덤프가 중단된다는 점에 유의하세요 . Enter 키를 눌러 활성화 여부를 선택합니다 .

- **Encrypt Swap?** - 스왑을 암호화할지 여부입니다 . 시스템이 부팅될 때마다 임시 키를 사용하여 스왑을 암호화하고 재부팅 시 이를 삭제합니다 . Enter 키를 눌러 활성화 여부를 선택합니다 . 스왑 암호화에 대한 자세한 내용은 '스왑 암호화하기 '를 참조하세요 .

T를 선택하여 Pool Type과 풀을 구성할 디스크를 구성합니다 .

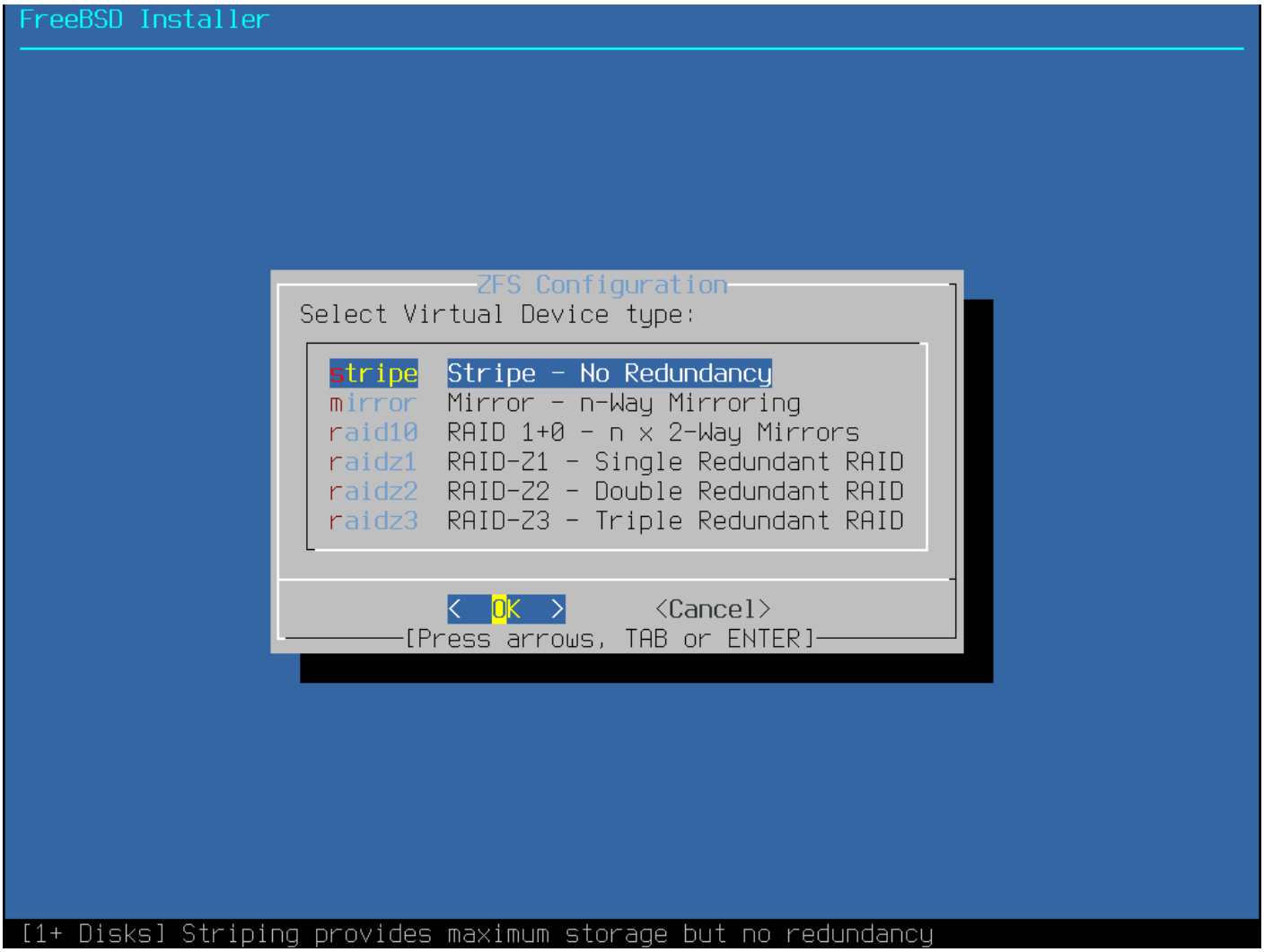
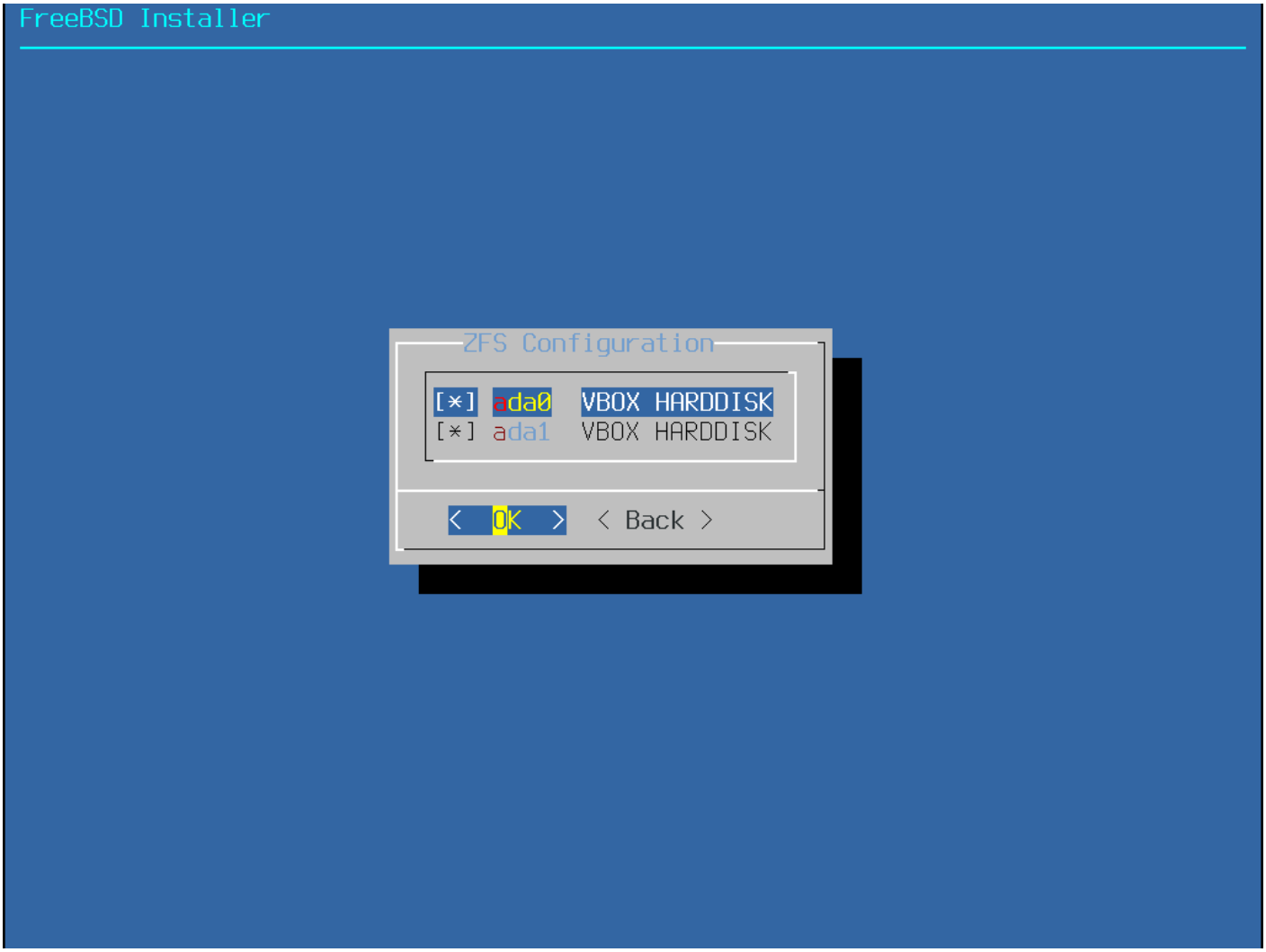


Figure 21. ZFS Pool Type

다음은 이 메뉴에서 선택할 수 있는 Pool Type에 대한 요약입니다 :

- stripe - 스트라이핑은 연결된 모든 장치에 최대 저장 공간을 제공하지만 중복성은 제공하지 않습니다 . 하나의 디스크에 장애가 발생하면 풀의 데이터는 돌이킬 수 없이 손실됩니다 .
- mirror - 미러링은 모든 디스크에 모든 데이터의 전체 사본을 저장합니다 . 미러링은 모든 디스크에서 데이터를 병렬로 읽기 때문에 읽기 성능이 우수합니다 . 풀의 모든 디스크에 데이터를 써야 하므로 쓰기 성능은 느립니다 . 하나의 디스크를 제외한 모든 디스크에 장애가 발생할 수 있습니다 . 이 옵션을 사용하려면 디스크가 2개 이상 필요합니다 .
- raid10 - 스트라이프 미러 . 최고의 성능을 제공하지만 가장 적은 스토리지를 제공합니다 . 이 옵션을 사용하려면 짝수 이상의 디스크와 최소 4개의 디스크가 필요합니다 .
- raidz1 - 단일 중복 RAID. 하나의 디스크에 동시에 장애가 발생하도록 허용합니다 . 이 옵션을 사용하려면 디스크가 3개 이상 필요합니다 .
- raidz2 - 이중 중복 RAID. 두 개의 디스크에 동시에 장애가 발생하도록 허용합니다 . 이 옵션을 사용하려면 디스크가 4개 이상 필요합니다 .



- raidz3 - 삼중 중복 RAID. 세 개의 디스크에 동시에 장애가 발생할 수 있습니다 . 이 옵션을 사용하려면 디스크가 5개 이상 필요합니다 .

Pool Type을 선택하면 사용 가능한 디스크 목록이 표시되고 풀을 구성할 디스크를 하나 이상 선택하라는 메시지가 표시됩니다 . 그런 다음 충분한 디스크가 선택되었는지 확인하기 위해 구성의 유효성을 검사합니다 . 유효성 검사에 실패하면 **<Change Selection>**을 선택하여 디스크 목록으로 돌아가거나 **<Back>**를 선택하여 Pool Type을 변경합니다 .

Figure 22. Disk Selection

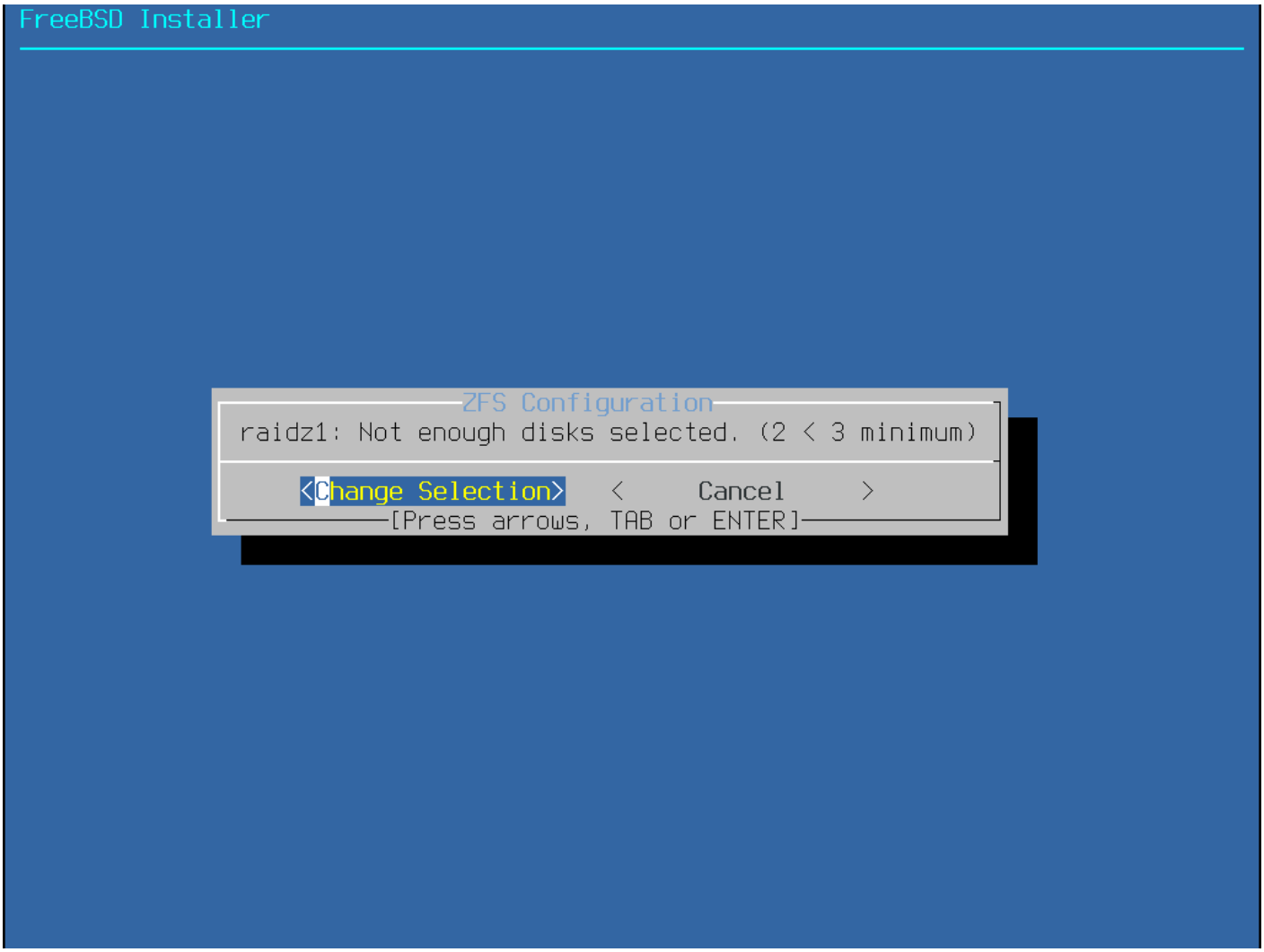


Figure 23. Invalid Selection

목록에서 하나 이상의 디스크가 누락되었거나 설치 프로그램이 시작된 후에 디스크가 연결된 경우 - **Rescan Devices**를 선택하여 사용 가능한 디스크 목록을 다시 채웁니다 .

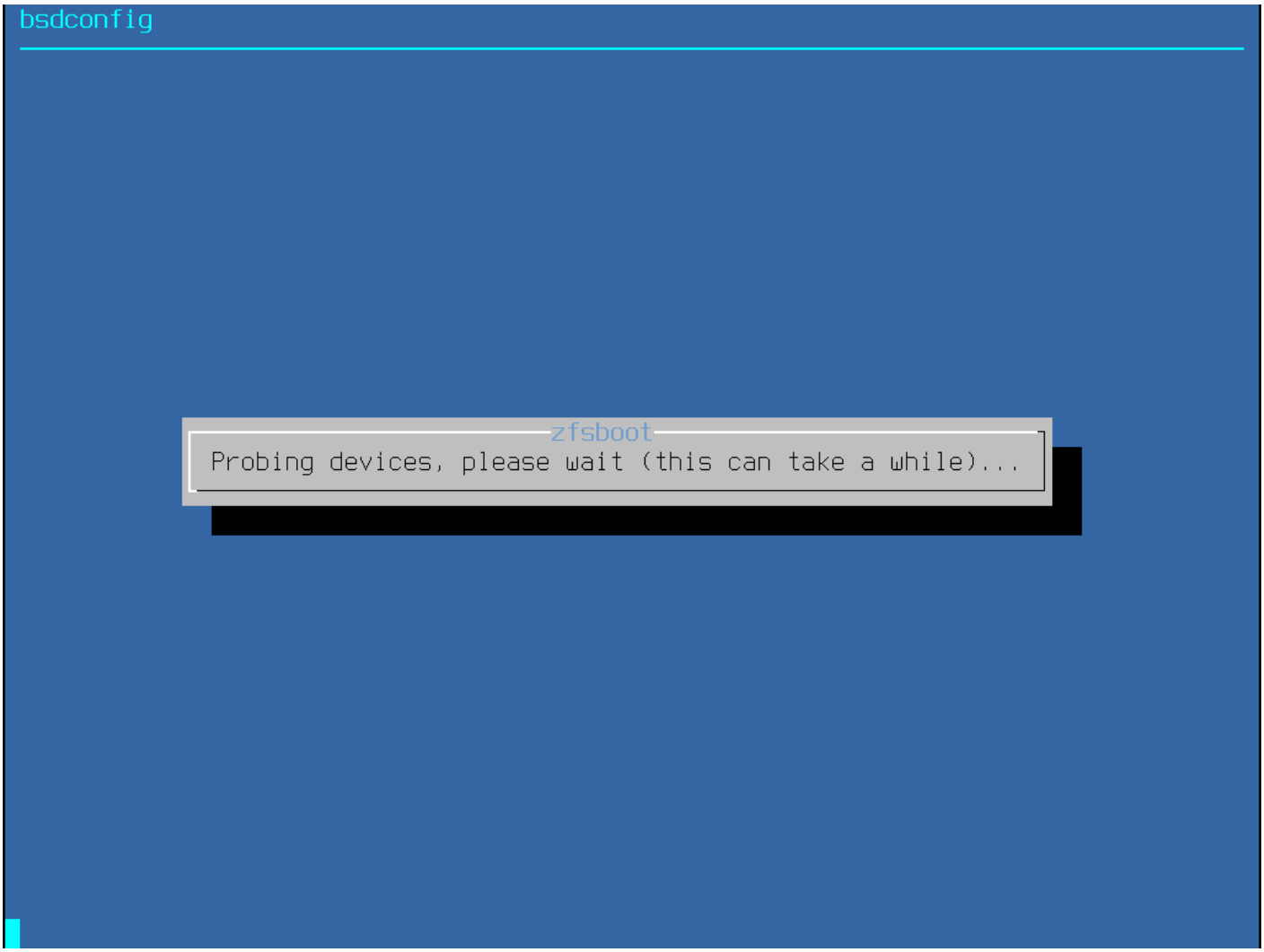


Figure 24. Rescan Devices

실수로 잘못된 디스크를 지우는 것을 방지하기 위해 - **Disk Info** 메뉴를 사용하여 각 디스크의 파티션 테이블과 장치 모델 번호 및 일련 번호와 같은 다양한 기타 정보 (가능한 경우)를 확인할 수 있습니다 .

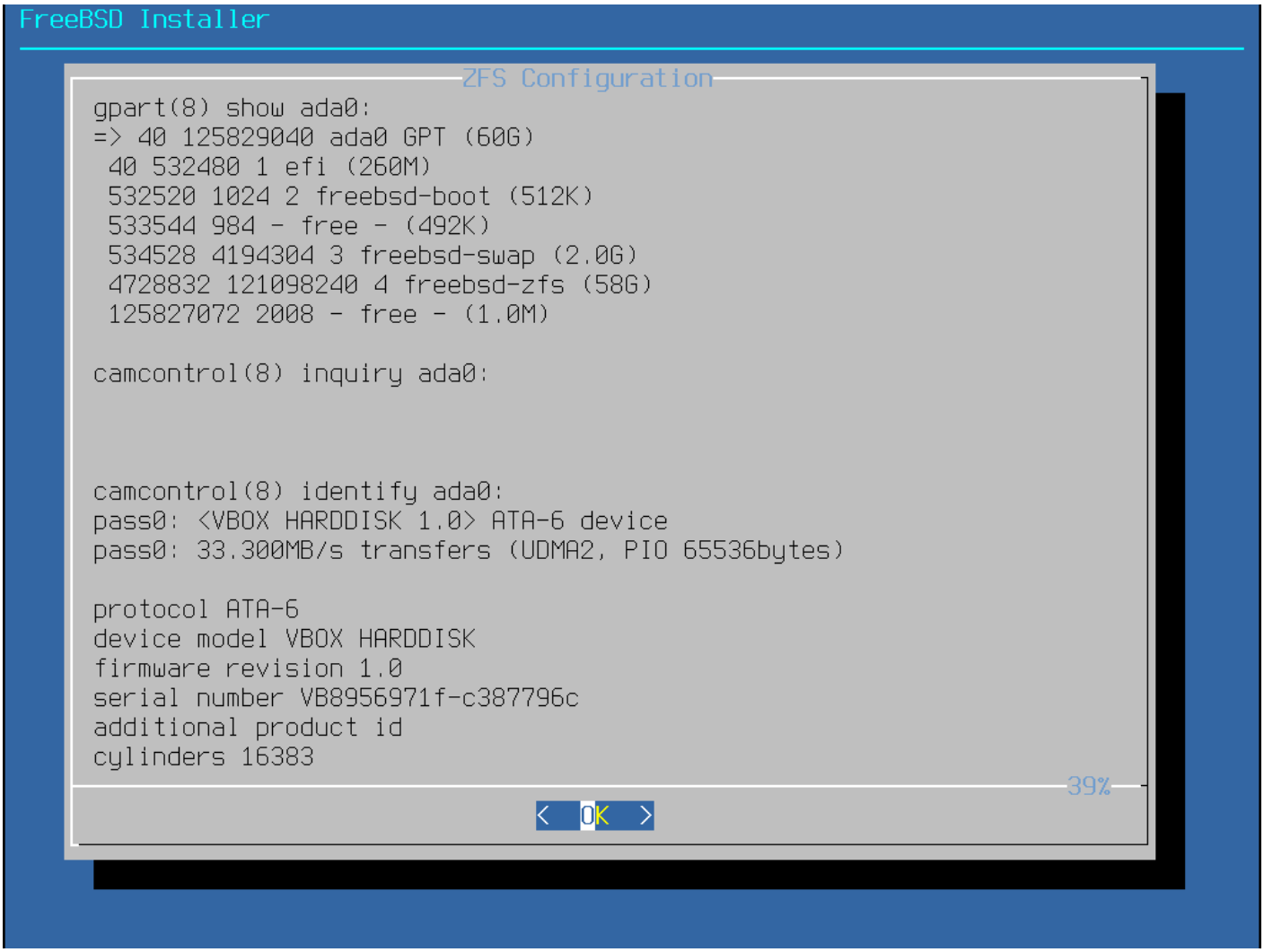


Figure 25. Analyzing a Disk

N을 선택하여 Pool Name을 구성합니다 . 원하는 이름을 입력한 다음 **<OK>**를 선택하여 설정하거나 **<Cancel>**을 선택하여 주 메뉴로 돌아가 기본 이름을 그대로 둡니다 .

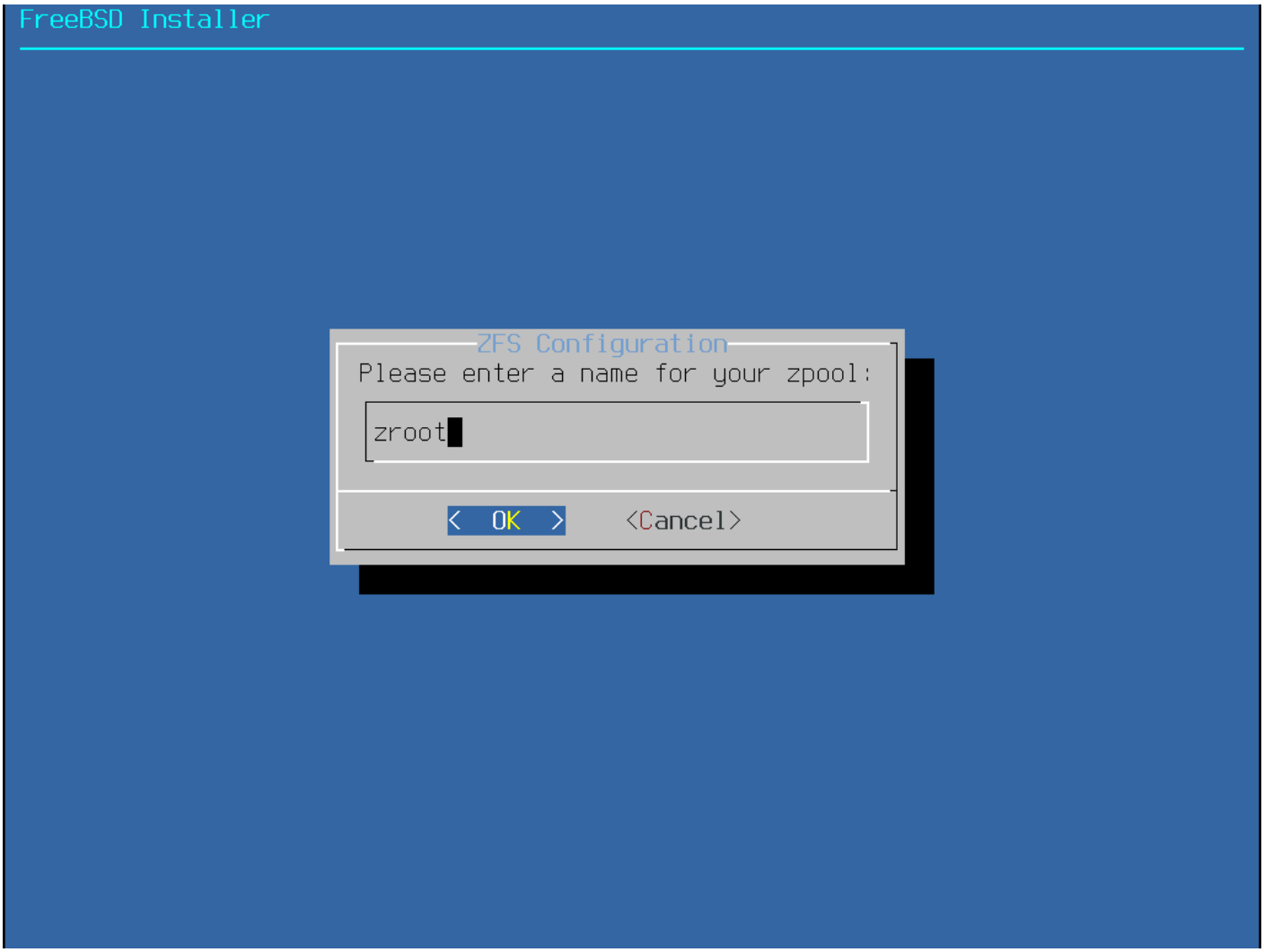


Figure 26. Pool Name

스왑 용량을 설정하려면 **S**를 선택합니다 . 원하는 스왑 용량을 입력한 다음 **<OK>**를 선택하여 설정하거나 **<Cancel>**을 선택하여 메인 메뉴로 돌아가 기본 용량을 그대로 둡니다 .

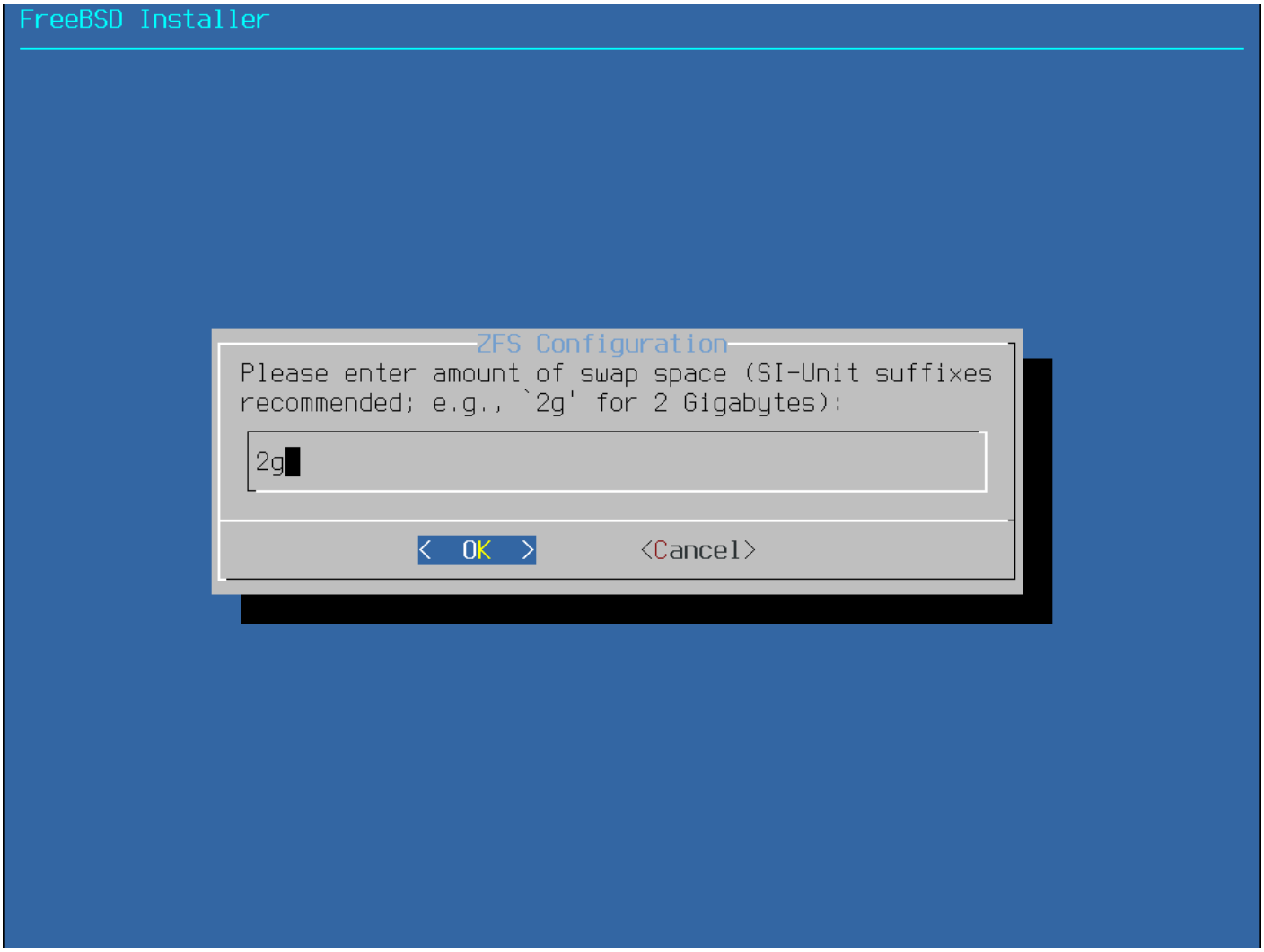


Figure 27. Swap Amount

모든 옵션을 원하는 값으로 설정했으면 메뉴 상단에서 **>>> install** 옵션을 선택합니다. 그러면 설치 관리자가 선택한 드라이브의 콘텐츠가 파괴되어 ZFS 풀이 생성되기 전에 마지막으로 취소할 수 있는 기회를 제공합니다.

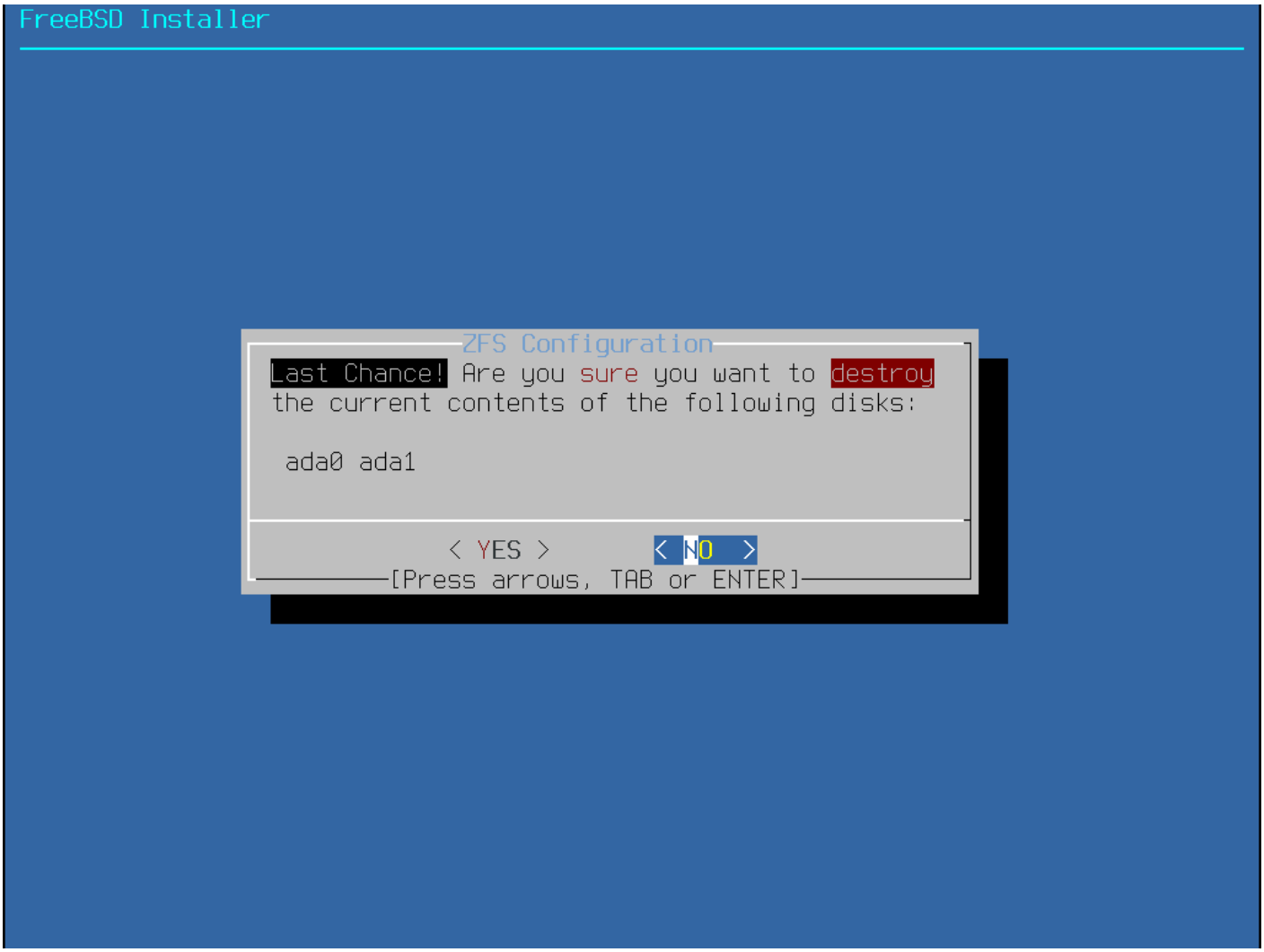


Figure 28. Last Chance

GELI [illegible] [illegible] [illegible]

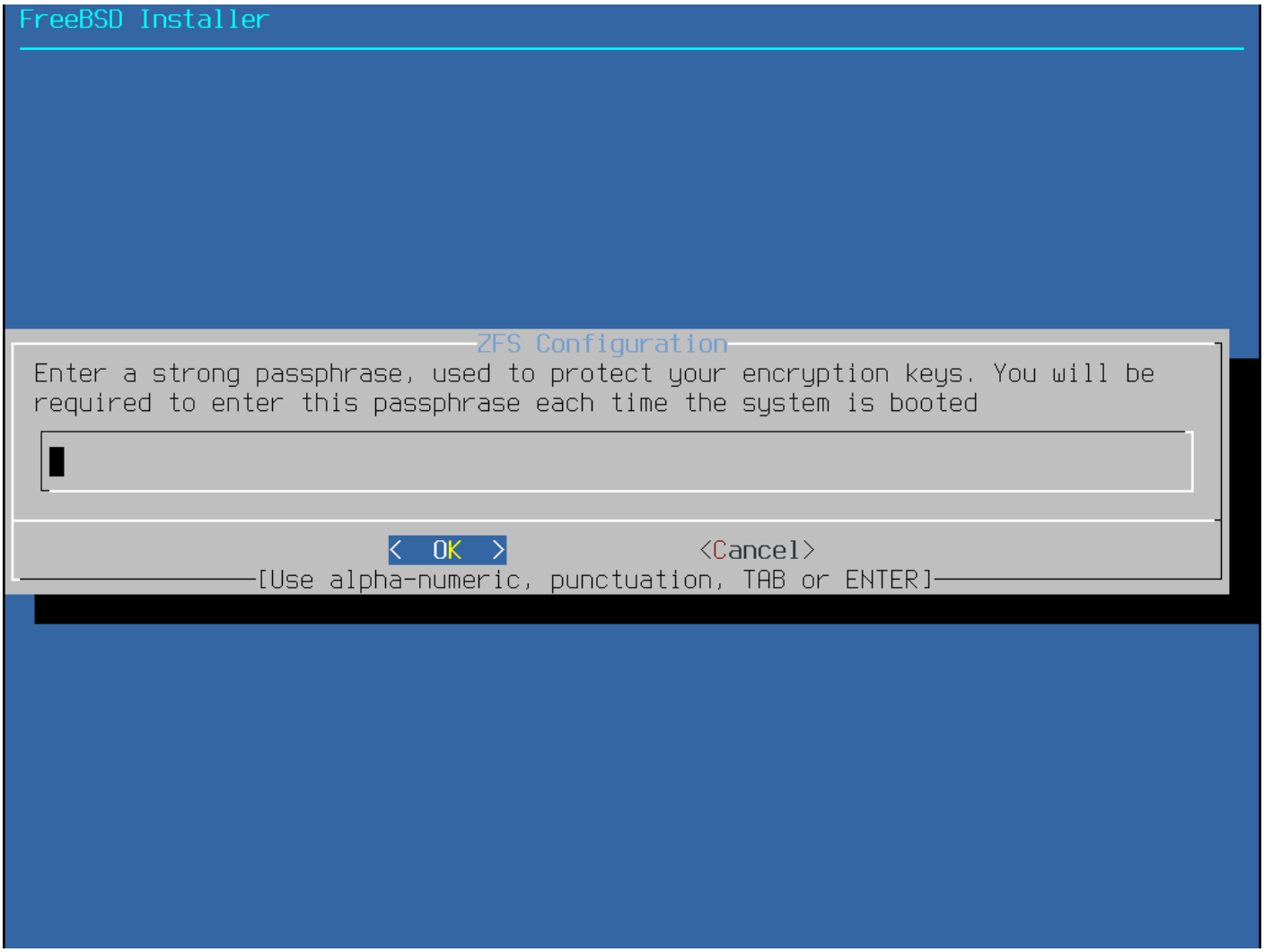


Figure 29. Disk Encryption Password

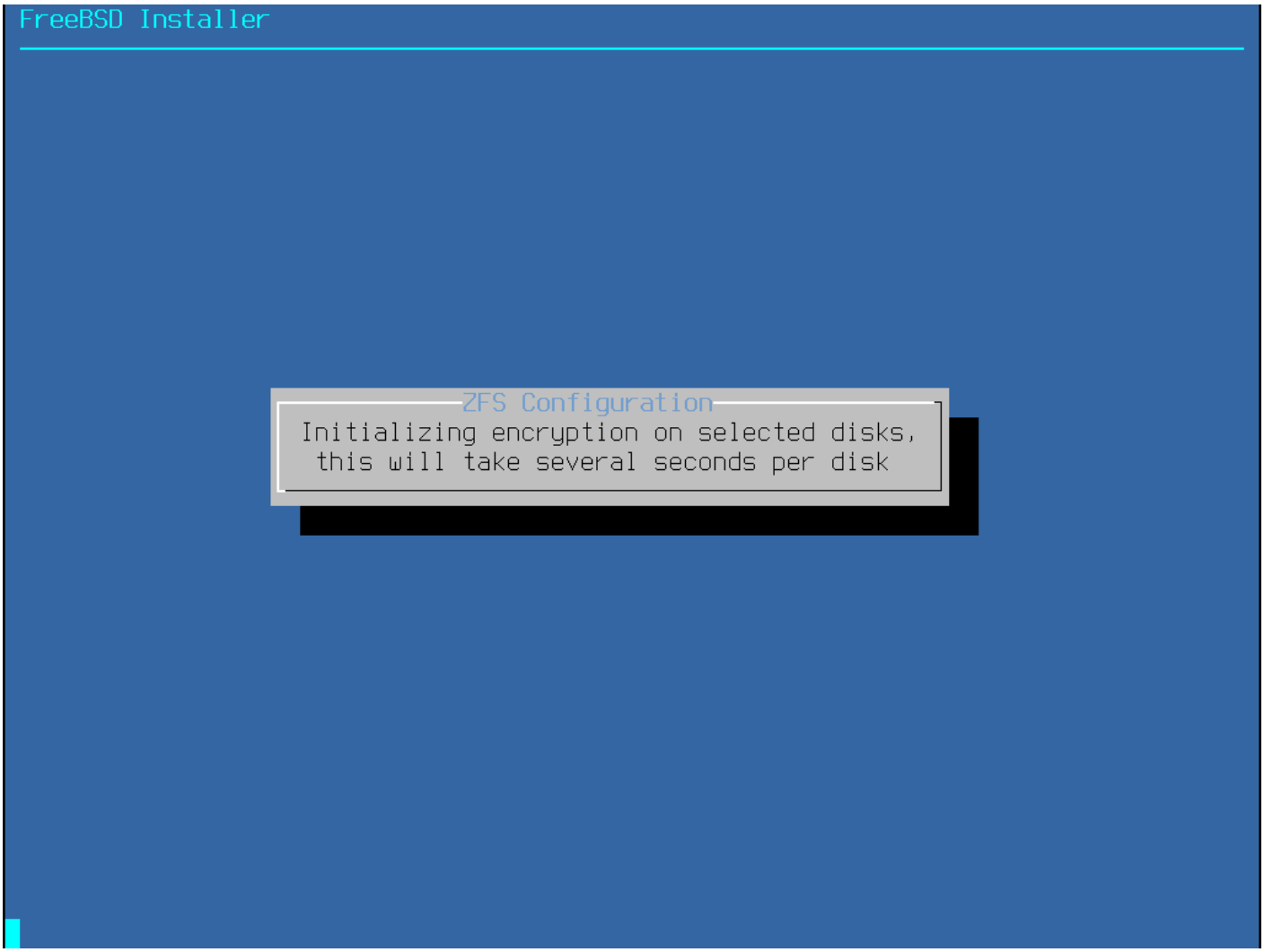


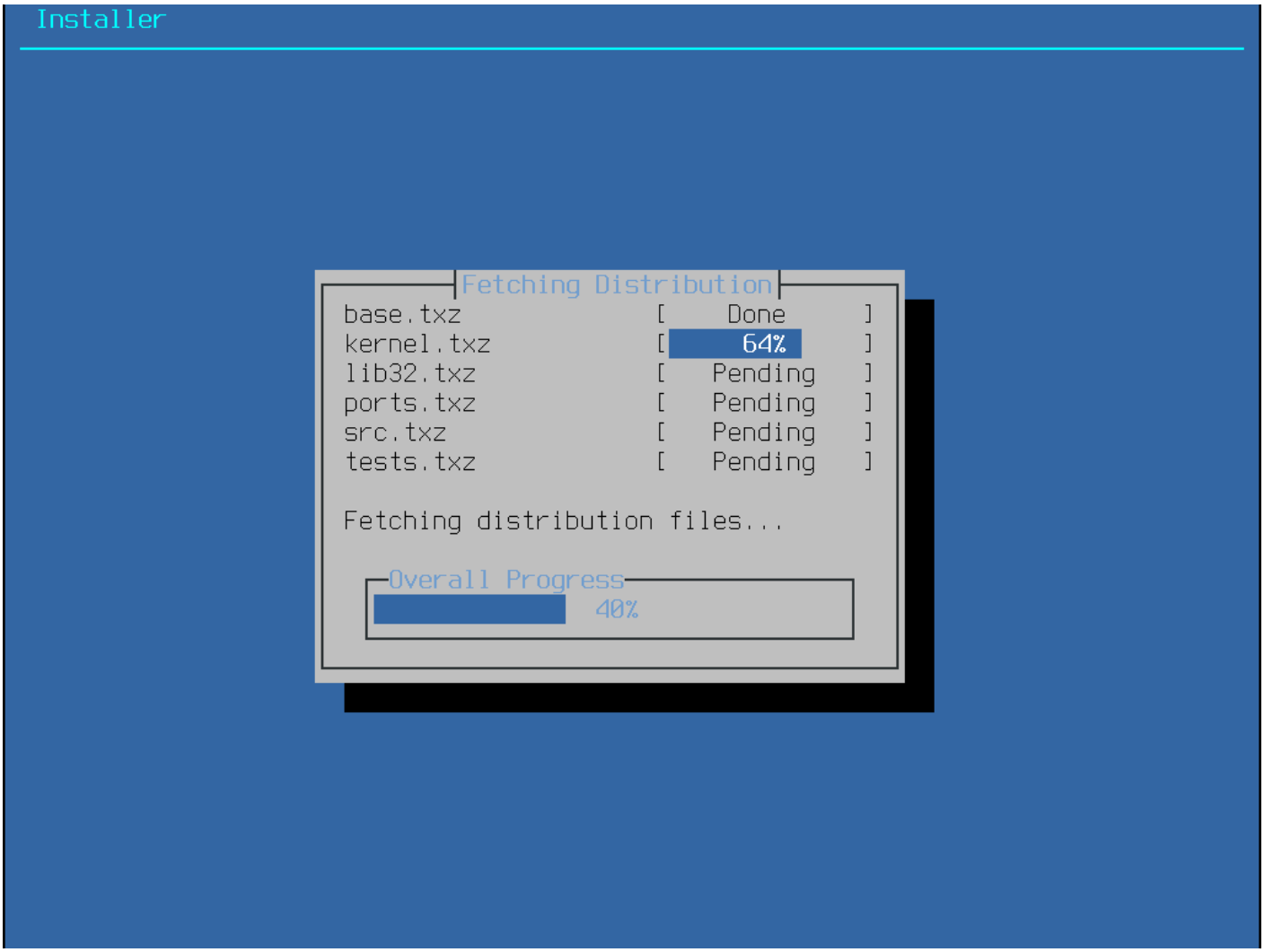
Figure 30. Initializing Encryption

그러면 설치가 정상적으로 진행됩니다 . 설치를 계속하려면 배포 파일 가져오기로 이동합니다 .

2.6.5. 셸 모드 파티셔닝

고급 설치를 만들 때 bsdinstall 파티셔닝 메뉴는 필요한 수준의 유연성을 제공하지 못할 수 있습니다 . 고급 사용자는 파티셔닝 메뉴에서 **Shell** 옵션을 선택하여 드라이브를 수동으로 파티션하고 , 파일 시스템을 만들고 , /tmp/bsdinstall_etc/fstab을 채우고 , /mnt 아래에 파일 시스템을 마운트할 수 있습니다 . 이 작업이 완료되면 exit를 입력하여 bsdinstall으로 돌아가 설치를 계속합니다 .

2.7. 배포 파일 가져오기



설치 시간은 선택한 배포판 , 설치 미디어 및 컴퓨터 속도에 따라 달라집니다 . 일련의 메시지가 진행 상황을 알려줍니다 .

먼저 설치 프로그램이 선택한 디스크를 포맷하고 파티션을 초기화합니다 . 그런 다음 `bootonly media` 또는 `mini memstick`의 경우 선택한 구성 요소를 다운로드합니다 :

Figure 31. Fetching Distribution Files

다음으로 배포 파일의 무결성을 확인하여 다운로드 중에 파일이 손상되었거나 설치 미디어에서 잘못 읽히지 않았는지 확인합니다 :

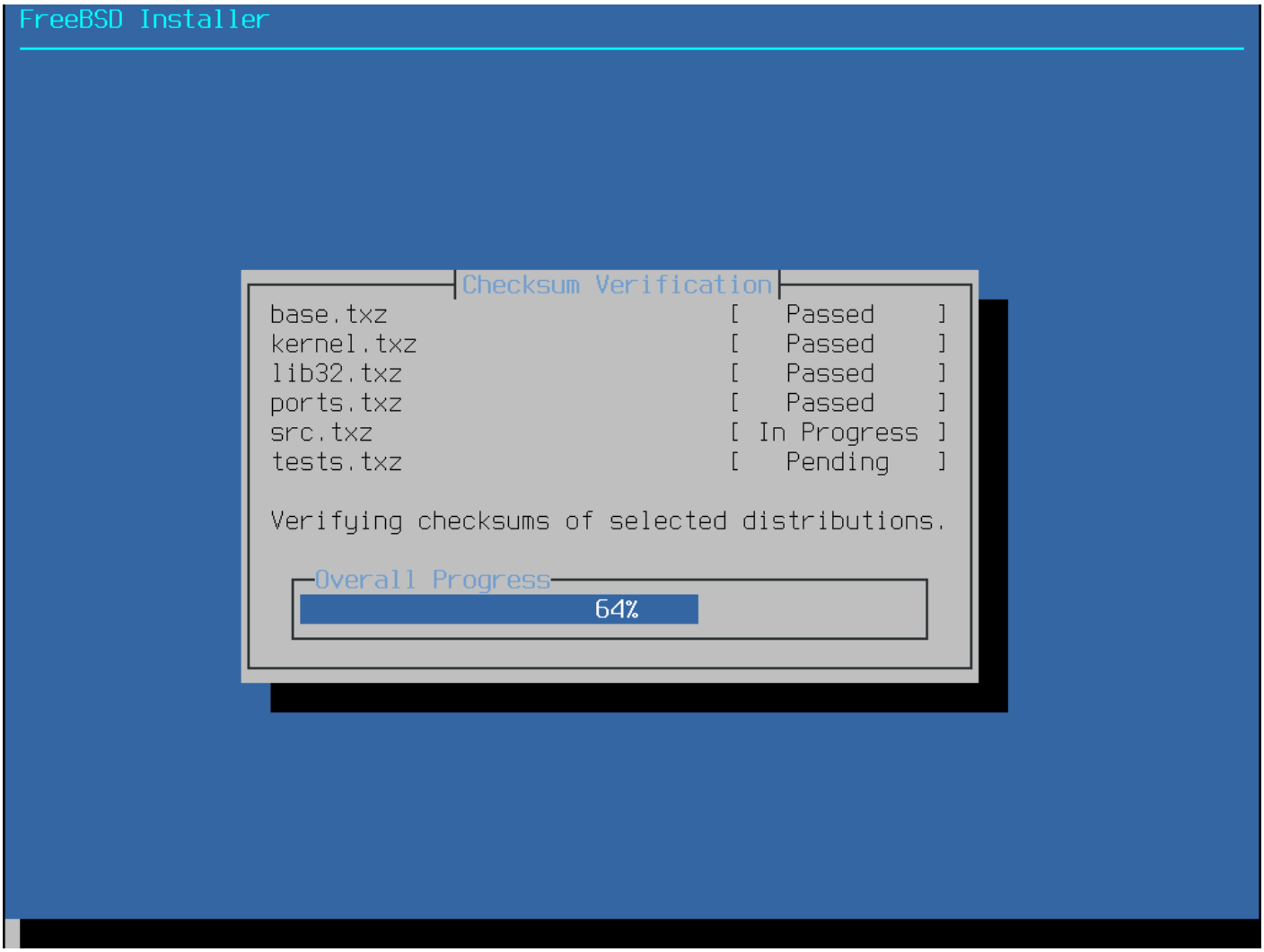


Figure 32. Verifying Distribution Files

마지막으로 확인된 배포 파일이 디스크에 추출됩니다 :

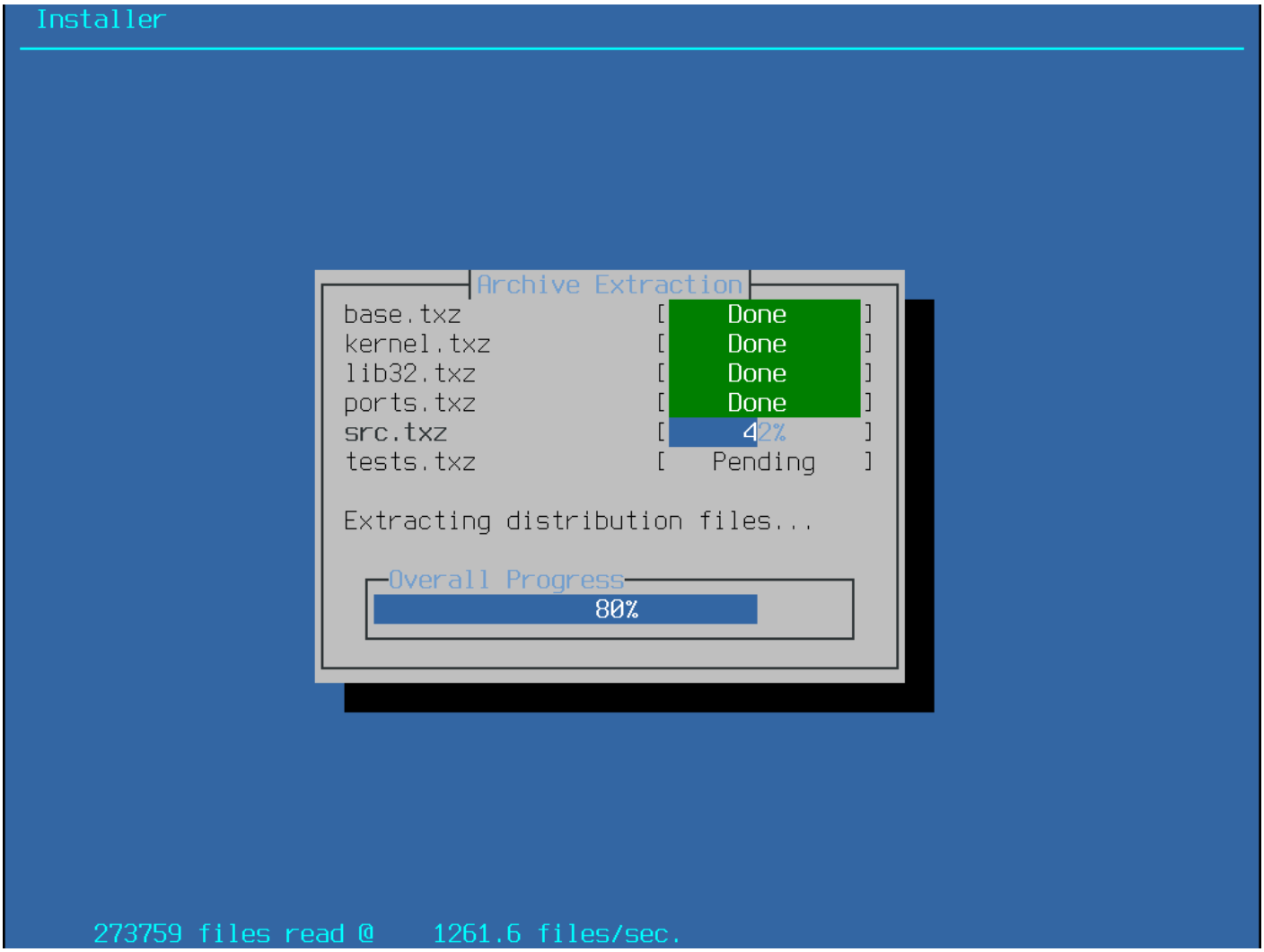


Figure 33. Extracting Distribution Files

bsdinstall .

2.8.

2.8.1. root

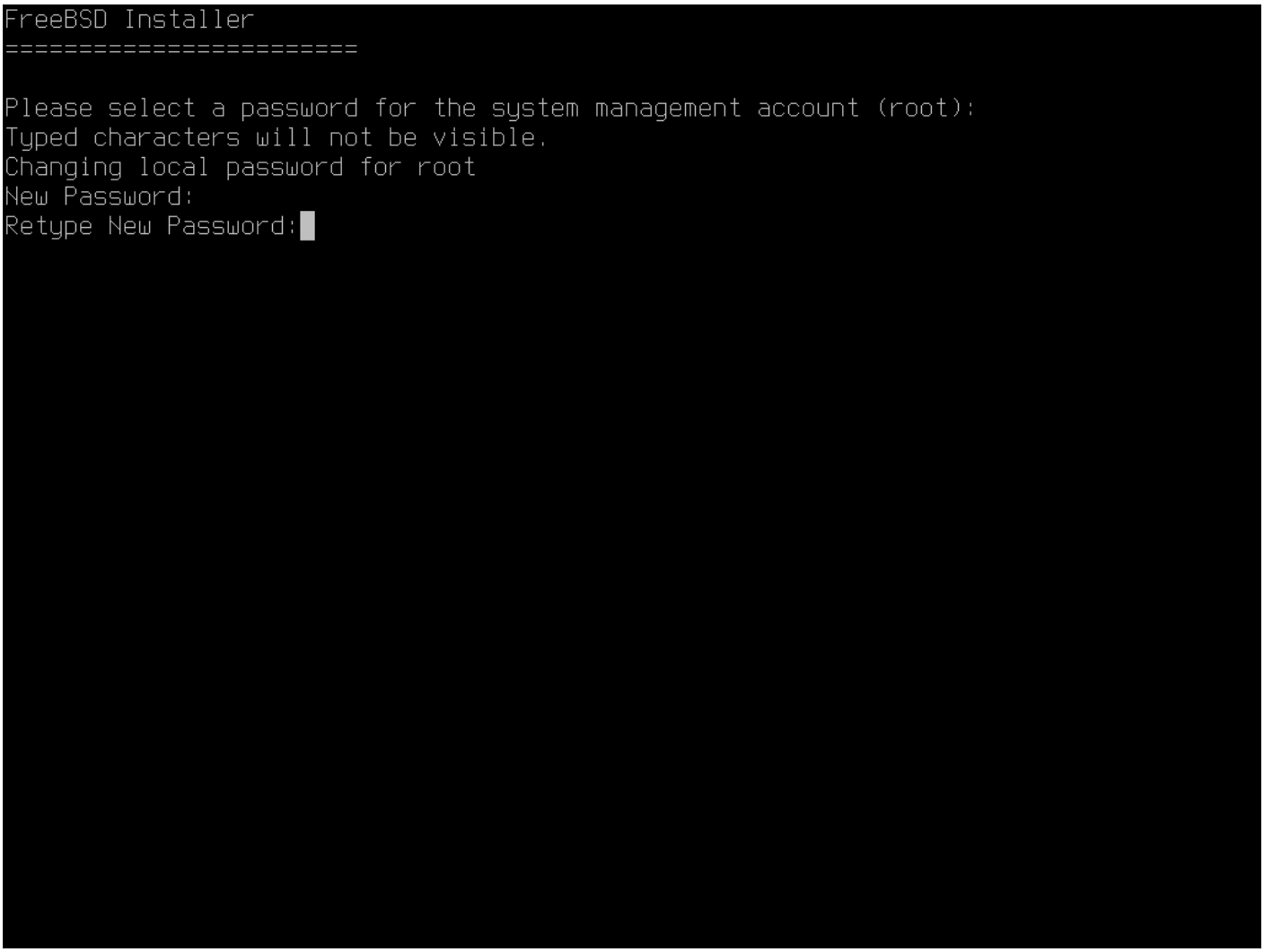


Figure 34. Setting the root Password

2.8.2. 네트워크 인터페이스 구성

다음으로 컴퓨터에서 찾을 수 있는 네트워크 인터페이스 목록이 표시됩니다 . 구성할 인터페이스를 선택합니다 .

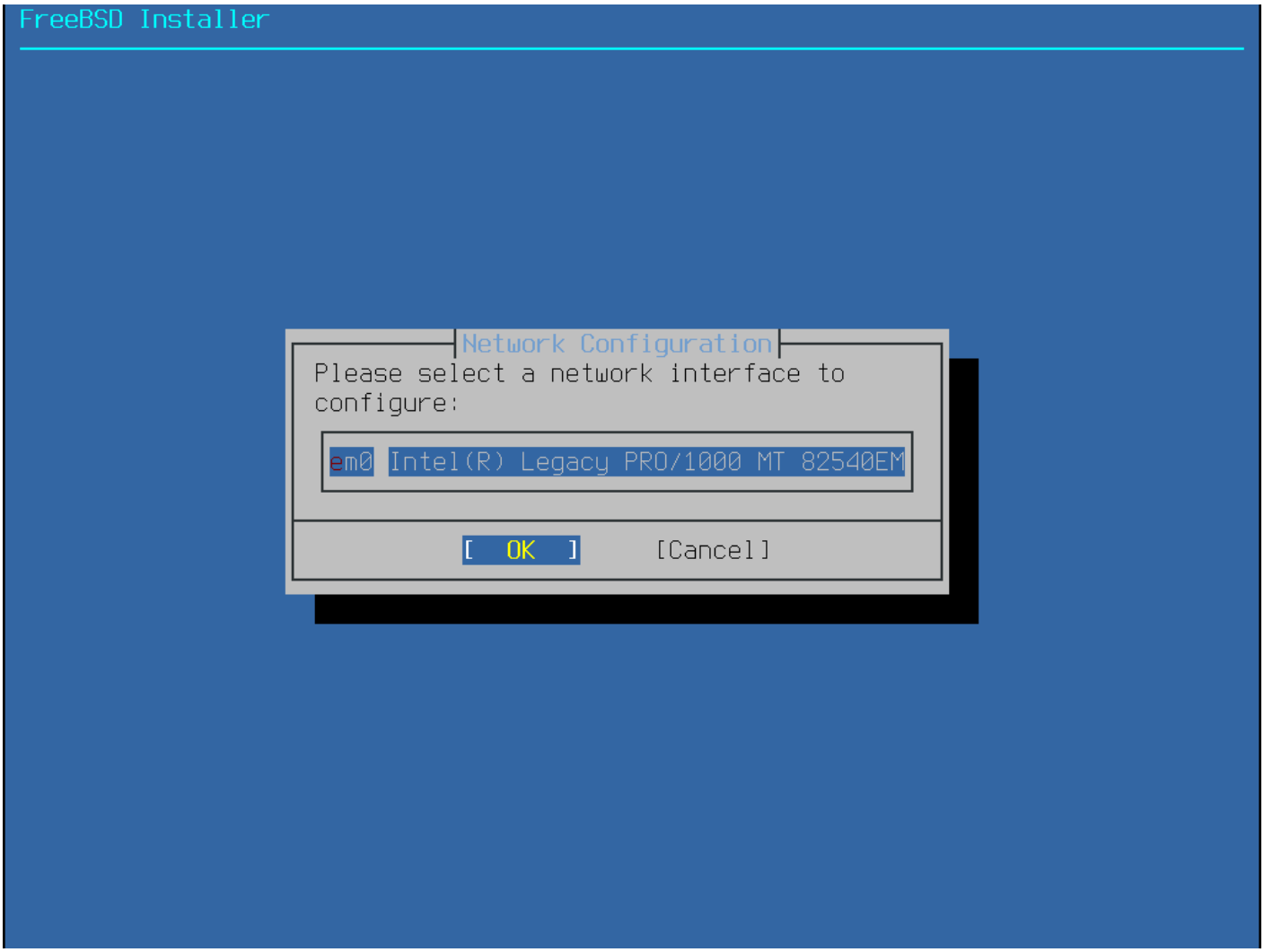


Figure 35. Choose a Network Interface

이더넷 인터페이스를 선택한 경우 , 설치 프로그램은 IPv4 네트워킹 선택에 표시된 메뉴로 건너뜁니다 . 무선 네트워크 인터페이스를 선택하면 시스템이 대신 무선 액세스 포인트를 검색합니다 :

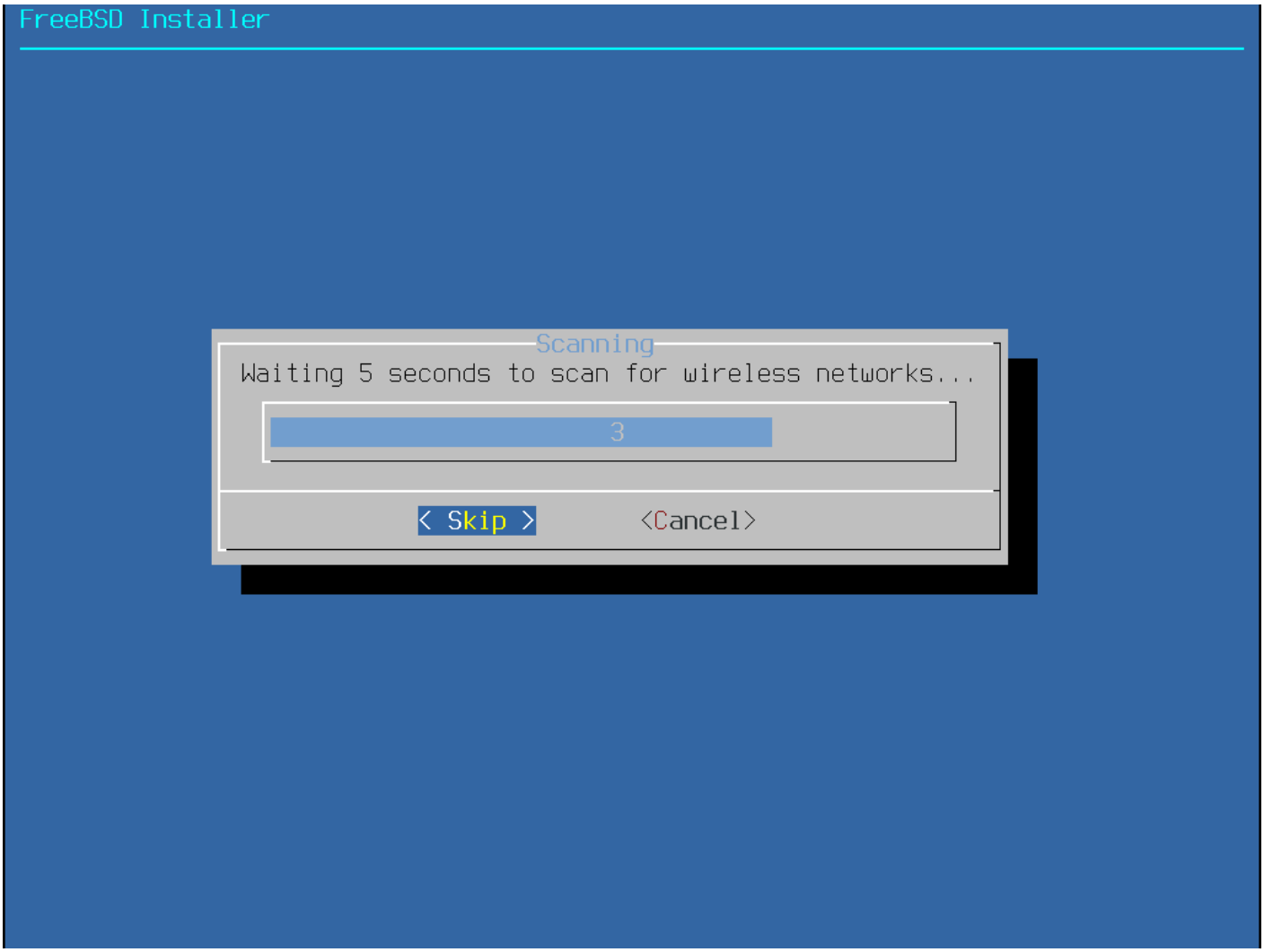


Figure 36. Scanning for Wireless Access Points

무선 네트워크는 각 네트워크에 부여된 짧고 고유한 이름인 SSID(서비스 세트 식별자)로 식별됩니다 . 스캔 중에 발견된 SSID가 나열되고 해당 네트워크에 사용할 수 있는 암호화 유형에 대한 설명이 이어집니다 . 원하는 SSID가 목록에 나타나지 않으면 **Rescan**을 선택하여 다시 스캔합니다 . 원하는 네트워크가 여전히 나타나지 않으면 안테나 연결에 문제가 있는지 확인하거나 컴퓨터를 액세스 포인트에 더 가까이 이동해 보세요 . 변경할 때마다 다시 스캔하세요 .

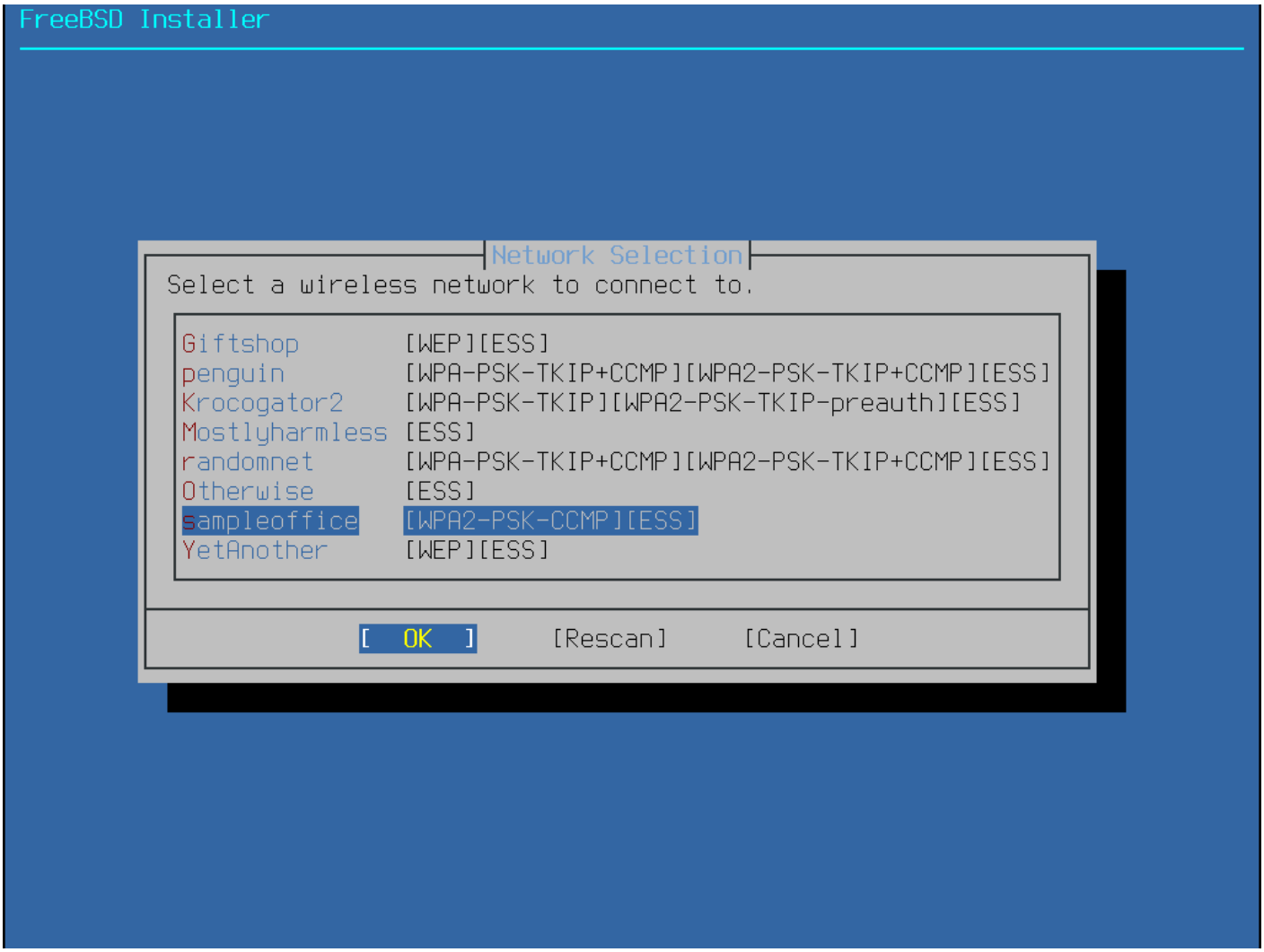


Figure 37. Choosing a Wireless Network

다음으로 선택한 무선 네트워크에 연결하기 위한 암호화 정보를 입력합니다 . 보안이 취약한 WEP와 같은 이전 암호화 유형보다 WPA2 암호화를 강력히 권장합니다 . 네트워크에서 WPA2를 사용하는 경우 PSK(사전 공유 키)라고도 하는 비밀번호를 입력합니다 . 보안상의 이유로 입력 상자에 입력한 문자는 별표로 표시됩니다 .

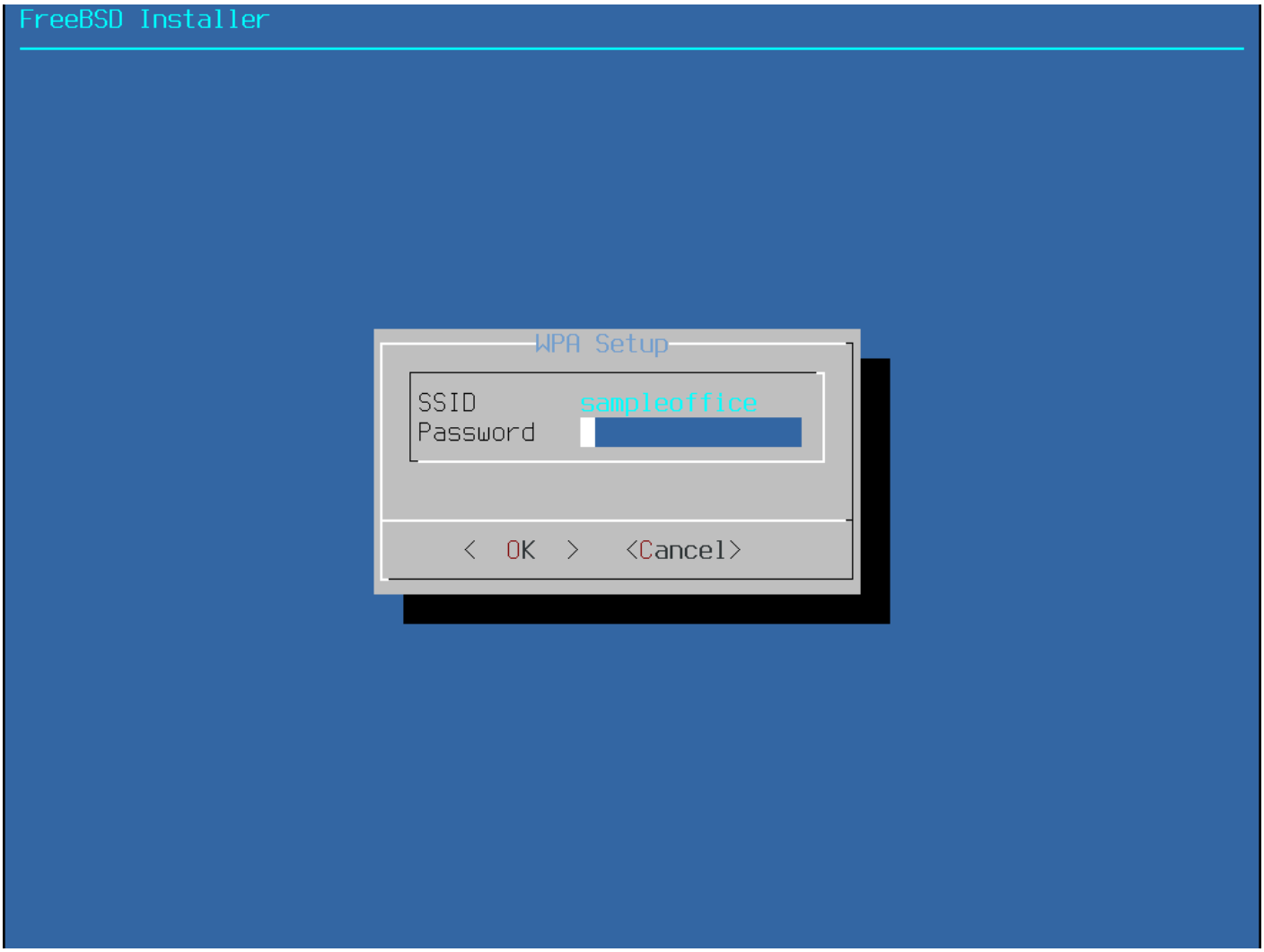


Figure 38. WPA2 Setup

다음으로 이더넷 또는 무선 인터페이스에 IPv4 주소를 구성할지 여부를 선택합니다 :

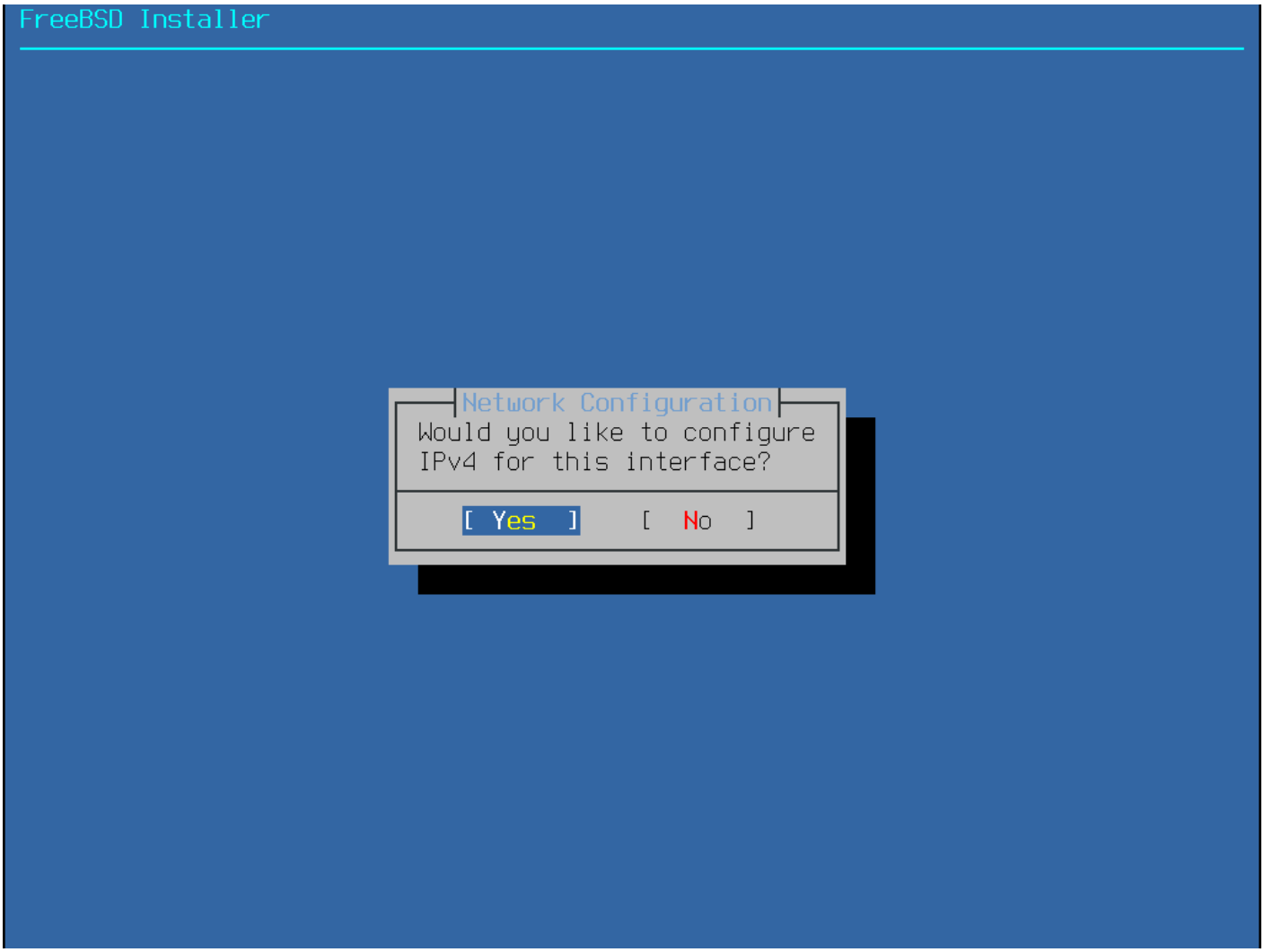


Figure 39. Choose IPv4 Networking

IPv4 구성 방법에는 두 가지가 있습니다 . DHCP는 네트워크 인터페이스를 자동으로 올바르게 구성하며 네트워크에서 DHCP 서버를 제공하는 경우 사용해야 합니다 . 그렇지 않으면 주소 지정 정보를 정적 구성으로 수동으로 입력해야 합니다 .

Note

임의의 네트워크 정보는 작동하지 않으므로 입력하지 마세요 . DHCP 서버를 사용할 수 없는 경우 네트워크 관리자 또는 인터넷 서비스 제공업체로부터 필수 네트워크 정보에 나열된 정보를 얻습니다 .

DHCP 서버를 사용할 수 있는 경우 다음 메뉴에서 **Yes**를 선택하여 네트워크 인터페이스를 자동으로 구성합니다 . 설치 프로그램이 DHCP 서버를 찾고 시스템의 주소 지정 정보를 가져오는 동안 잠시 멈추는 것처럼 보입니다 .

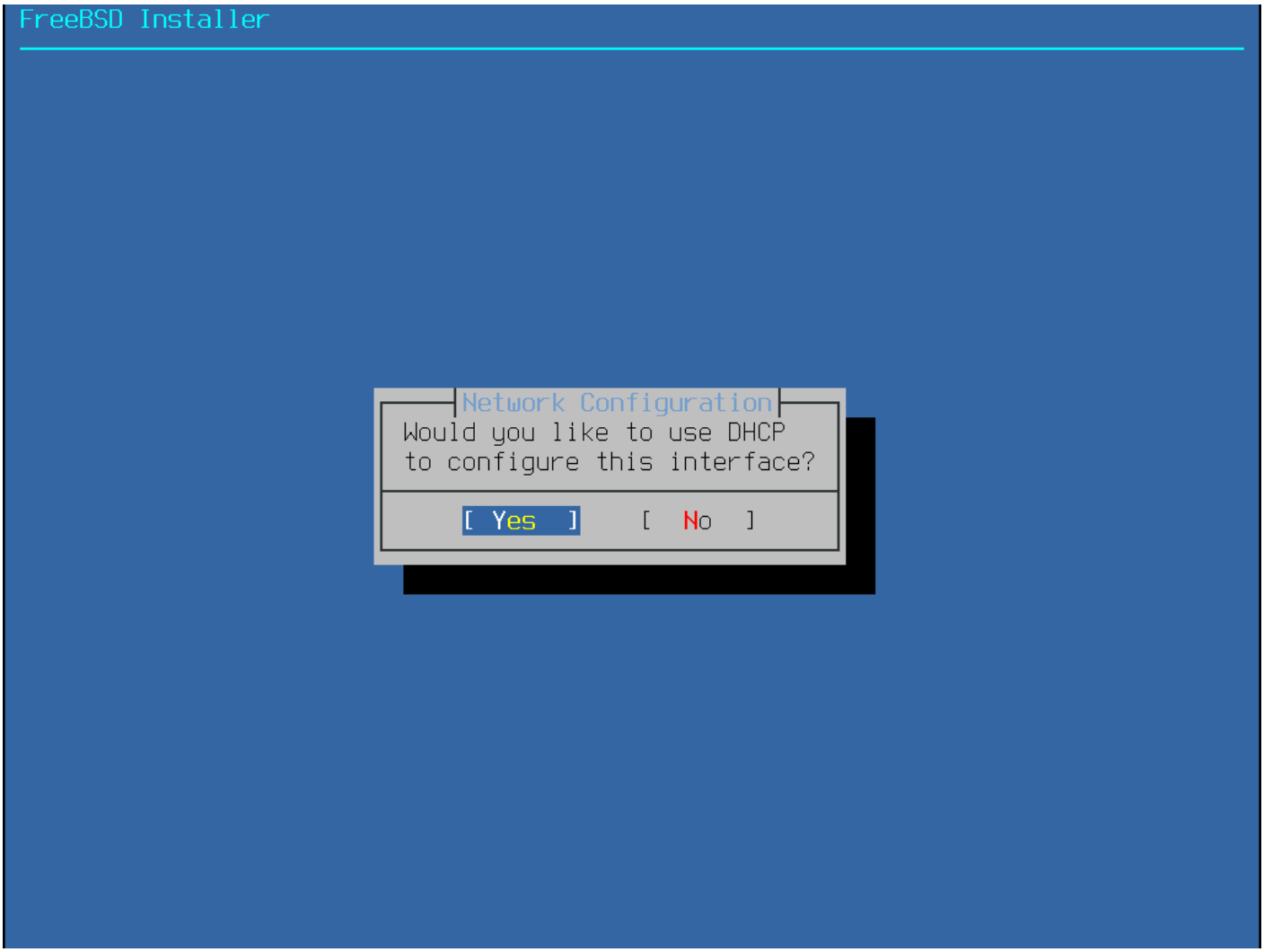


Figure 40. Choose IPv4 DHCP Configuration

DHCP 서버를 사용할 수 없는 경우 **No**를 선택하고 이 메뉴에 다음 주소 지정 정보를 입력합니다 :

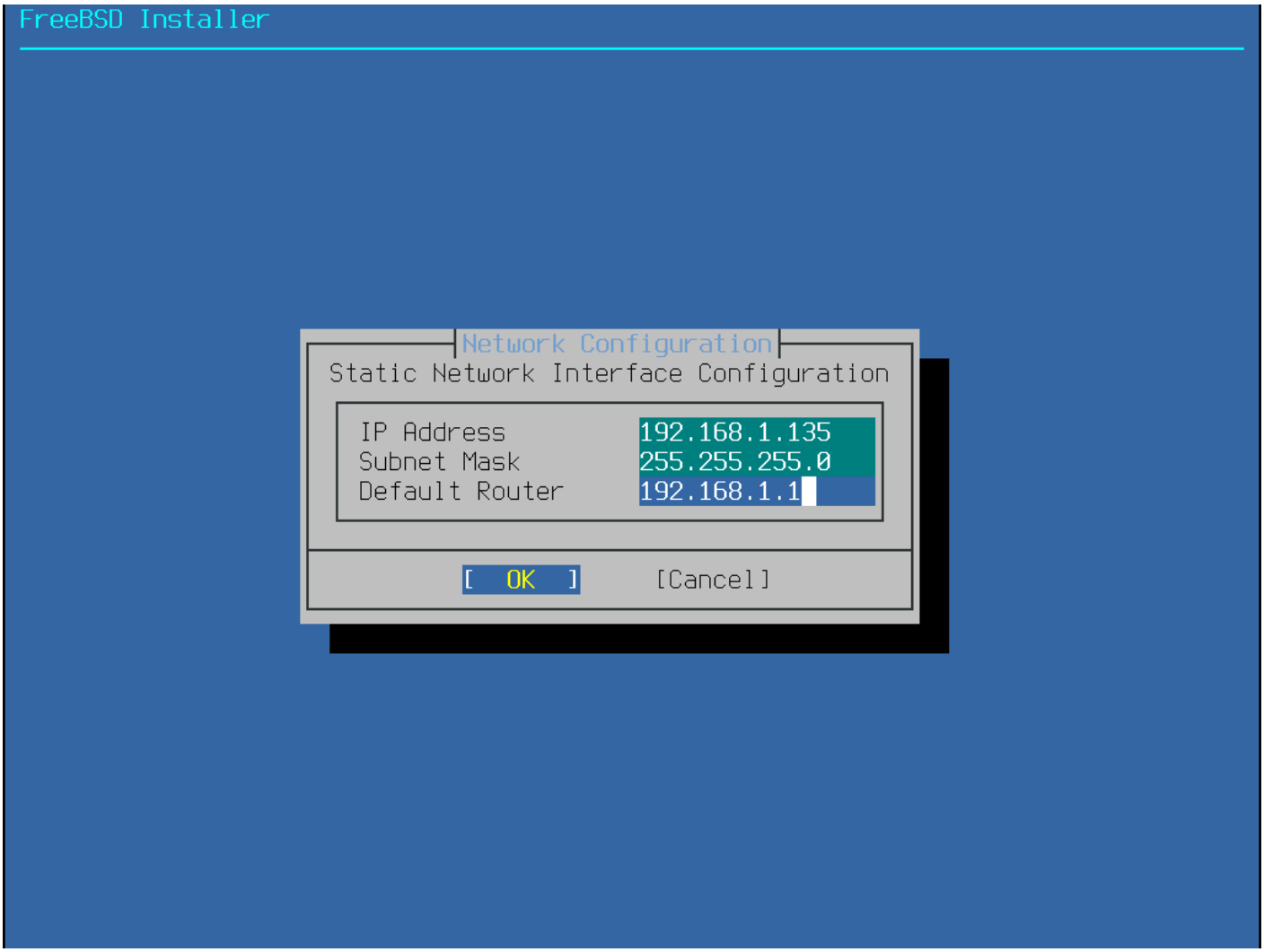


Figure 41. IPv4 Static Configuration

- IP Address - 이 컴퓨터에 할당된 IPv4 주소입니다 . 이 주소는 고유해야 하며 로컬 네트워크의 다른 장치에서 이미 사용 중이 아니어야 합니다 .
- Subnet Mask - 네트워크의 서브넷 마스크입니다 .
- Default Router - 네트워크의 기본 게이트웨이의 IP 주소입니다 .

다음 화면에서 인터페이스를 IPv6용으로 구성할지 여부를 묻습니다 . IPv6를 사용할 수 있고 원하는 경우 예를 선택하여 선택합니다 .

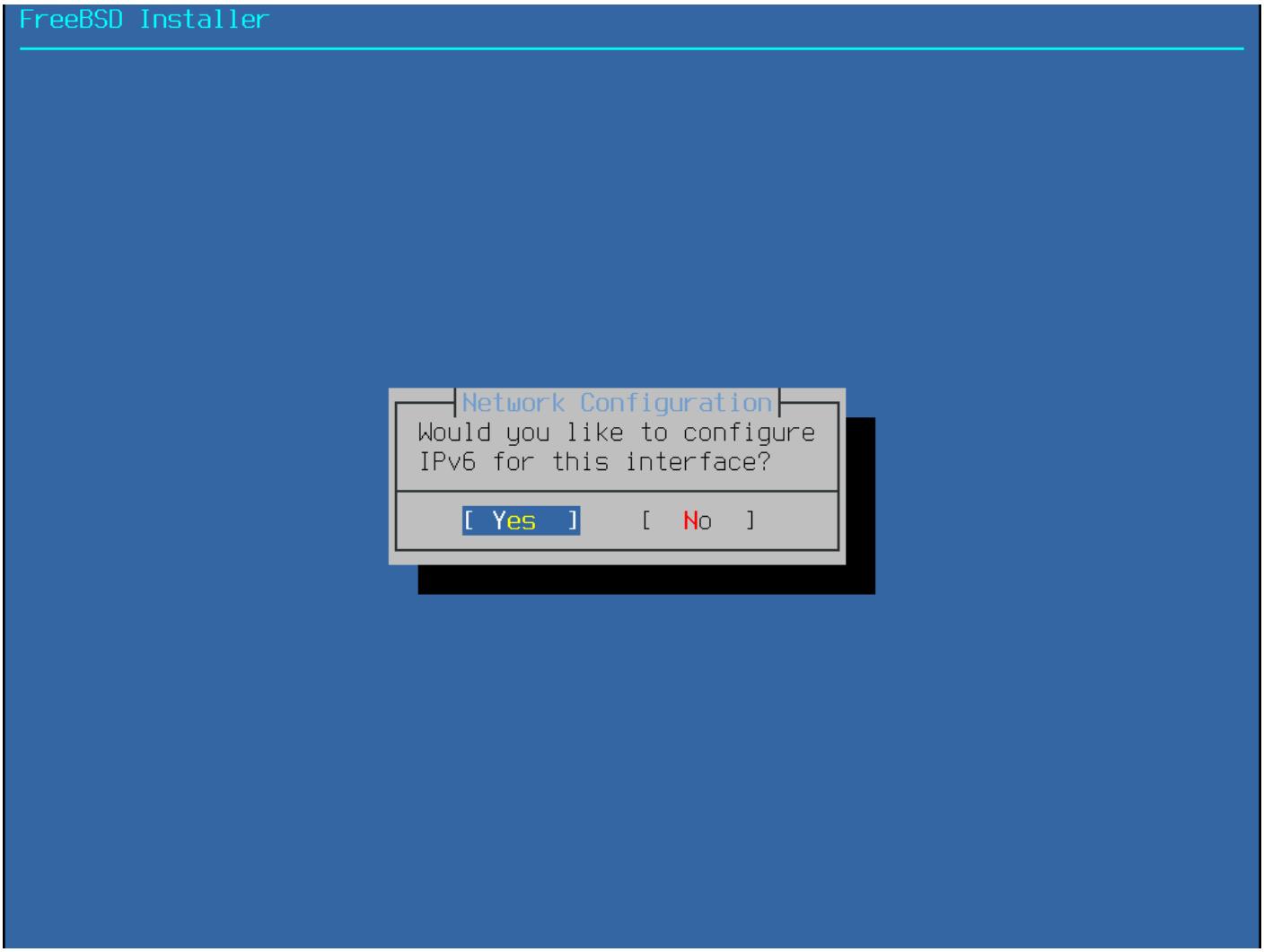


Figure 42. Choose IPv6 Networking

IPv6에는 두 가지 구성 방법도 있습니다 . SLAAC(StateLess Address AutoConfiguration)는 로컬 라우터에 올바른 구성 정보를 자동으로 요청합니다 . 자세한 내용은 rfc4862를 참조하세요 . 정적 구성은 네트워크 정보를 수동으로 입력해야 합니다 .

IPv6 라우터를 사용할 수 있는 경우 다음 메뉴에서 **Yes**를 선택하여 네트워크 인터페이스를 자동으로 구성합니다 . 라우터를 찾고 시스템의 주소 지정 정보를 얻는 동안 설치 프로그램이 1분 정도 일시 중지되는 것처럼 보입니다 .

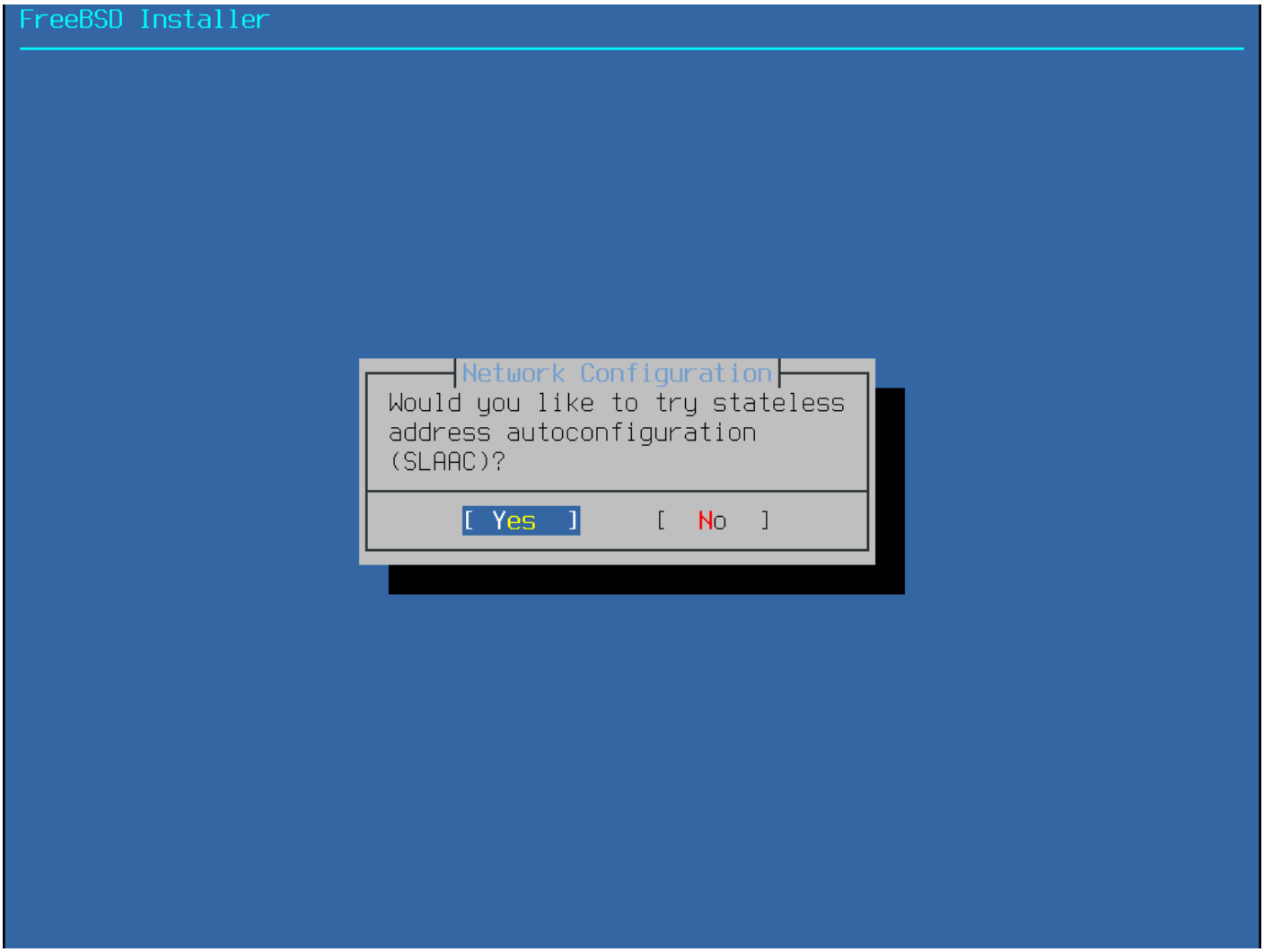


Figure 43. Choose IPv6 SLAAC Configuration

IPv6 라우터를 사용할 수 없는 경우 **No**를 선택하고 이 메뉴에 다음 주소 지정 정보를 입력합니다 :

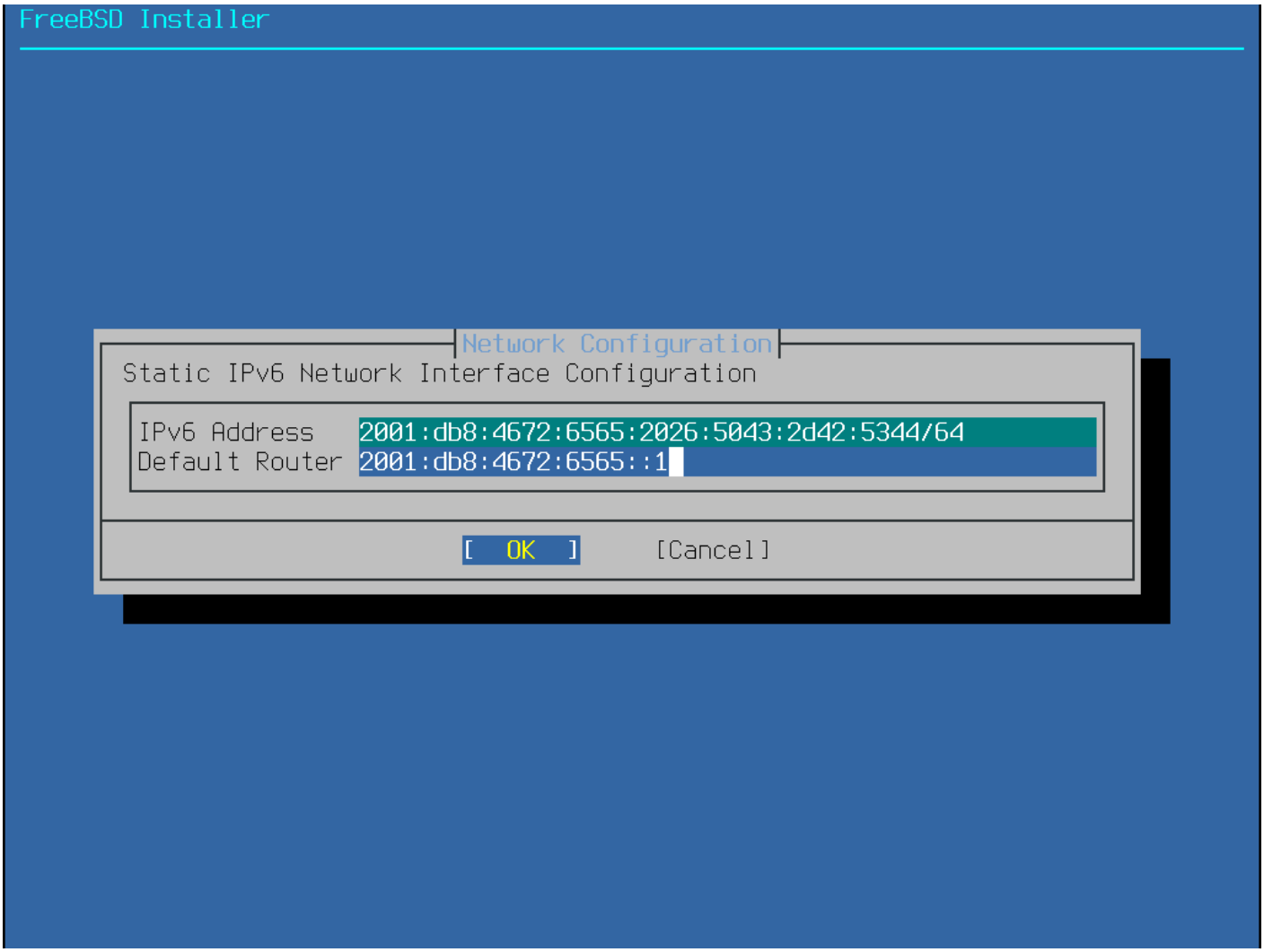


Figure 44. IPv6 Static Configuration

- IPv6 Address - 이 컴퓨터에 할당된 IPv6 주소입니다 . 이 주소는 고유해야 하며 로컬 네트워크의 다른 장치에서 이미 사용하고 있지 않아야 합니다 .
- Default Router - 네트워크의 기본 게이트웨이의 IPv6 주소입니다 .

마지막 네트워크 구성 메뉴는 호스트 이름을 네트워크 주소로 변환하는 DNS(도메인 이름 시스템) 확인자를 구성하는 데 사용됩니다 . 네트워크 인터페이스를 자동 구성하는 데 DHCP 또는 SLAAC를 사용했다면 Resolver Configuration 값이 이미 입력되어 있을 수 있습니다 . 그렇지 않은 경우 검색 필드에 로컬 네트워크의 도메인 이름을 입력합니다 . DNS #1 및 DNS #2는 DNS 서버의 IPv4 및 /또는 IPv6 주소입니다 . 하나 이상의 DNS 서버가 필요합니다 .

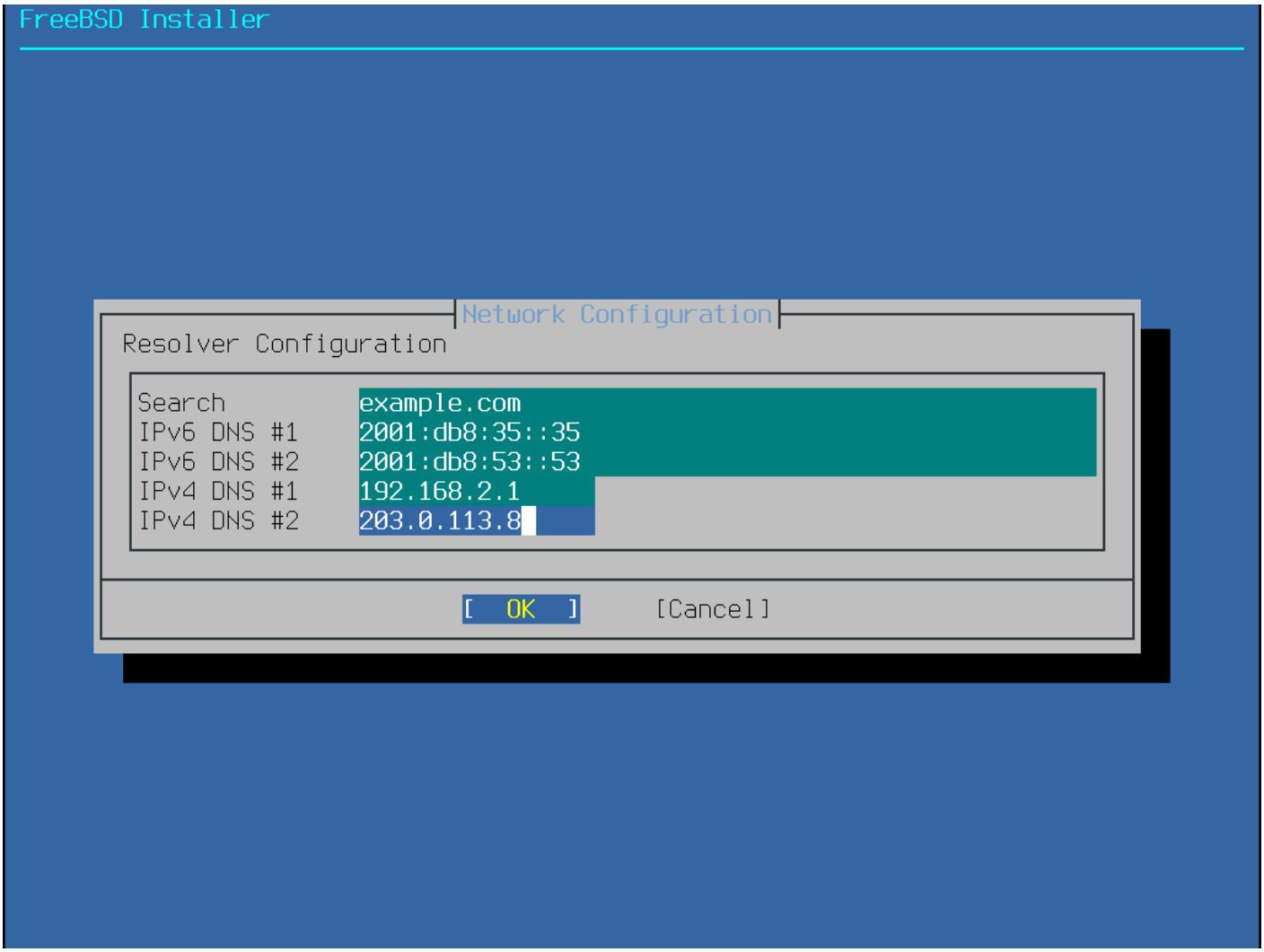


Figure 45. DNS Configuration

인터페이스가 구성되면 FreeBSD가 설치되는 컴퓨터와 같은 지역에 있는 미러 사이트를 선택합니다 . 미러가 대상 컴퓨터와 가까우면 파일을 더 빨리 검색할 수 있으므로 설치 시간이 단축됩니다 .

TIP

ftp://download.freebsd.org(메인 사이트)를 선택하면 가장 가까운 미러로 자동 라우팅됩니다 .

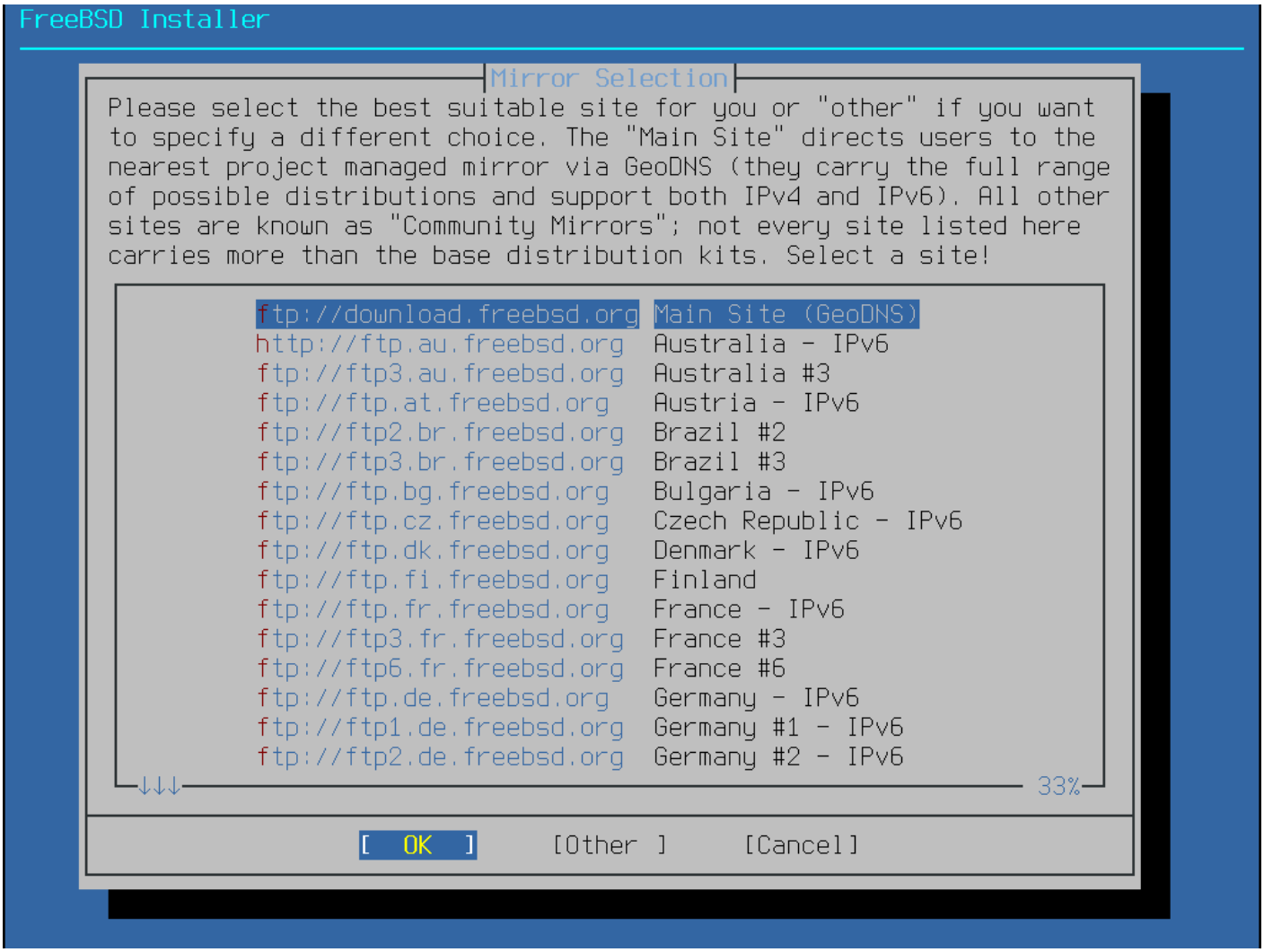


Figure 46. Choosing a Mirror

2.8.3. 표준 시간대 설정

다음 일련의 메뉴는 지역 , 국가 및 시간대를 선택하여 정확한 현지 시간을 결정하는 데 사용됩니다 . 시간대를 설정하면 시스템이 서머타임과 같은 지역 시간 변경 사항을 자동으로 수정하고 다른 시간대 관련 기능을 올바르게 수행할 수 있습니다 .

여기에 표시된 예는 유럽 스페인 본토 표준 시간대에 위치한 컴퓨터의 경우입니다 . 지리적 위치에 따라 선택 항목이 달라질 수 있습니다 .

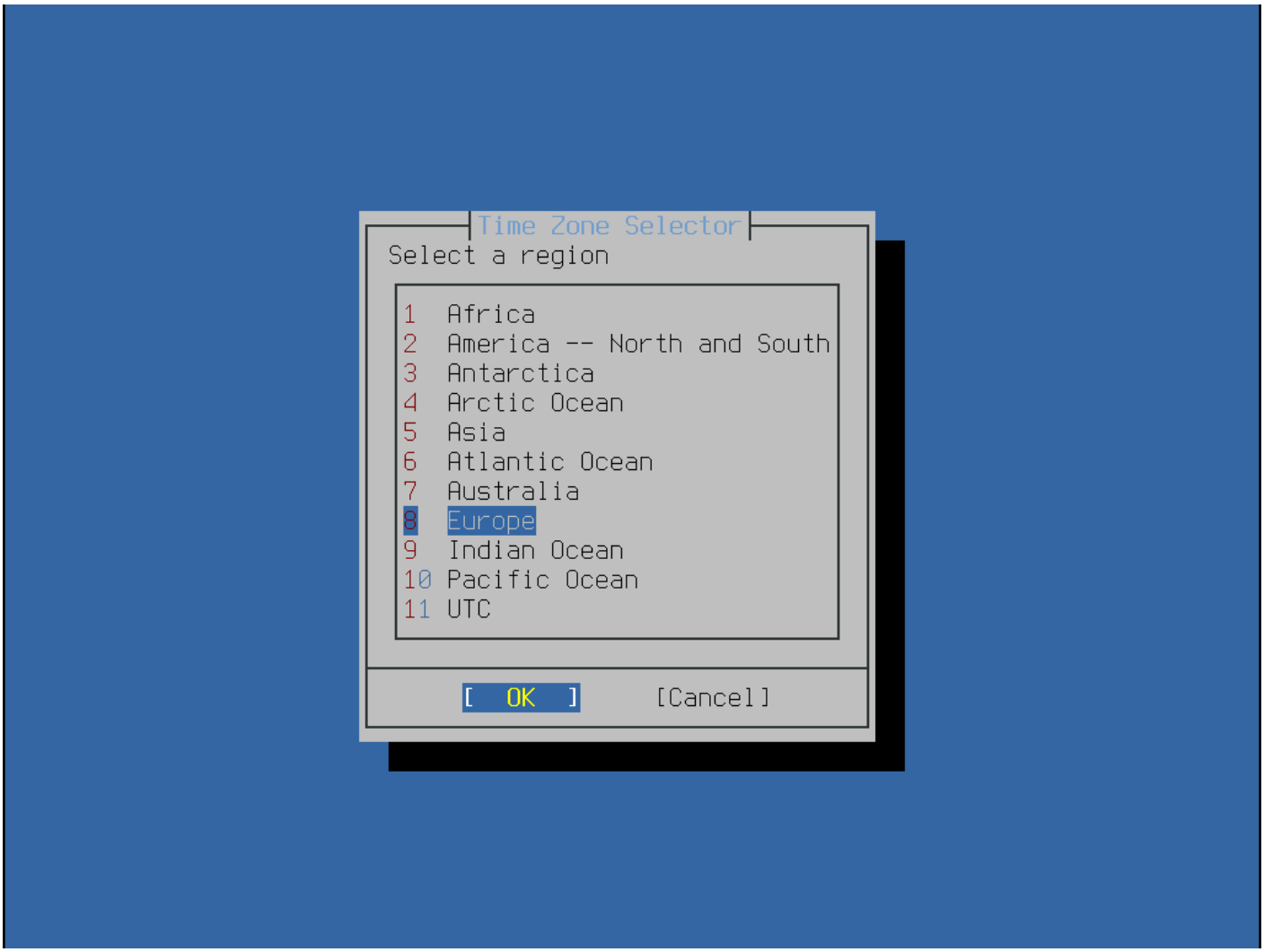


Figure 47. Select a Region

화살표 키를 사용하여 적절한 영역을 선택한 다음 **Enter** 키를 누릅니다 .

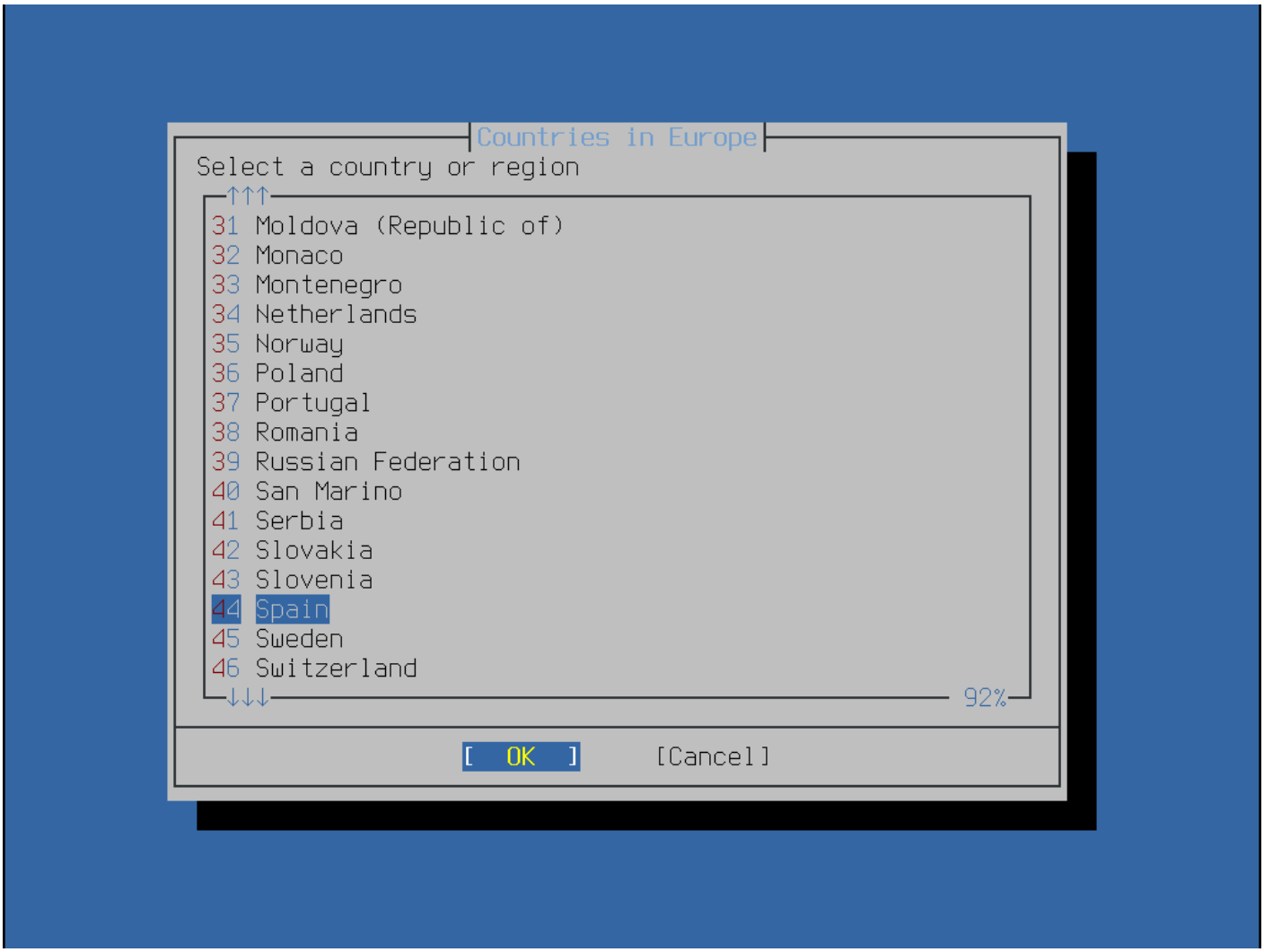


Figure 48. Select a Country

화살표 키를 사용하여 해당 국가를 선택하고 **Enter** 키를 누릅니다 .

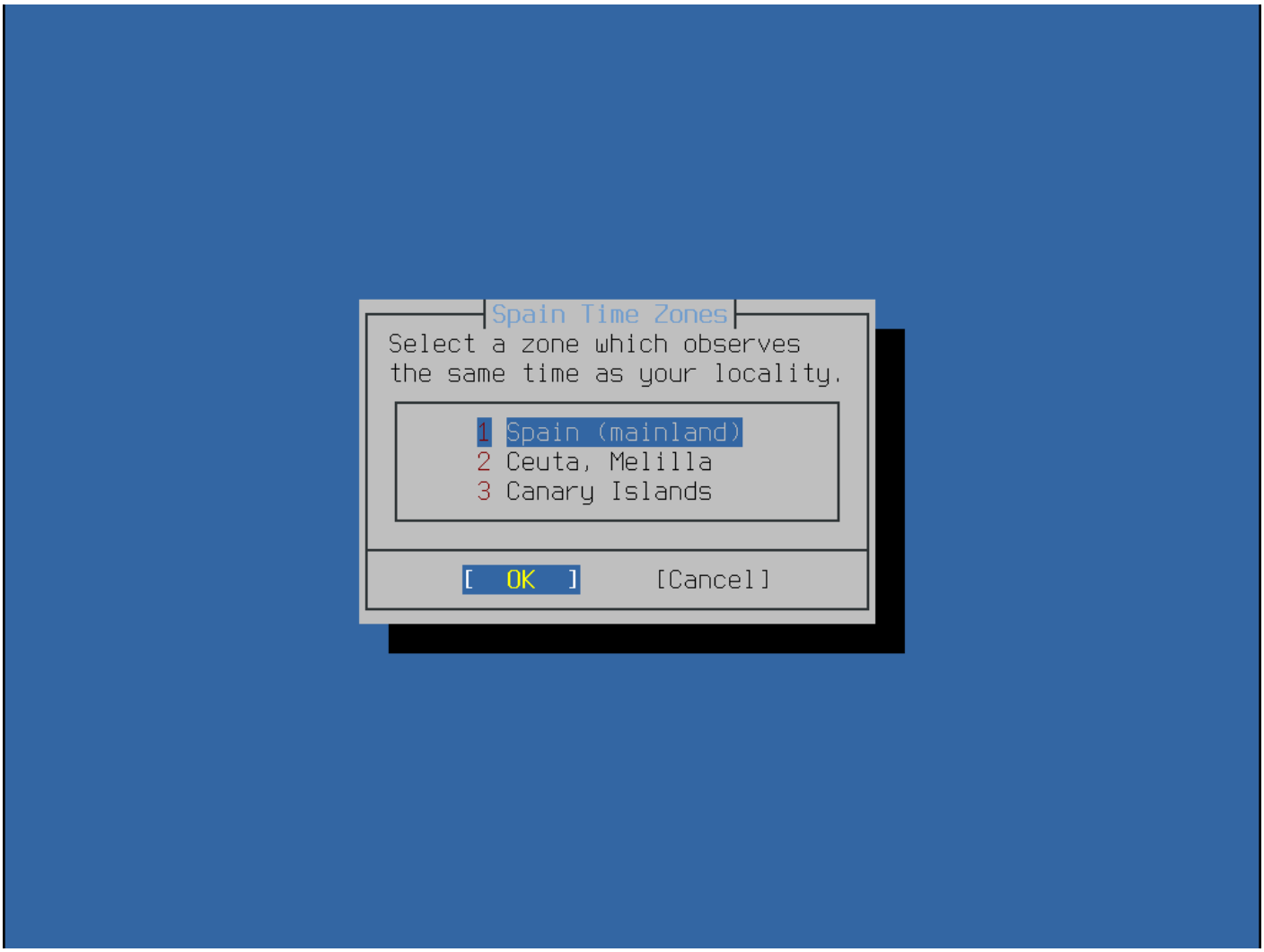


Figure 49. Select a Time Zone

화살표 키를 사용하여 적절한 시간대를 선택하고 **Enter** 키를 누릅니다 .

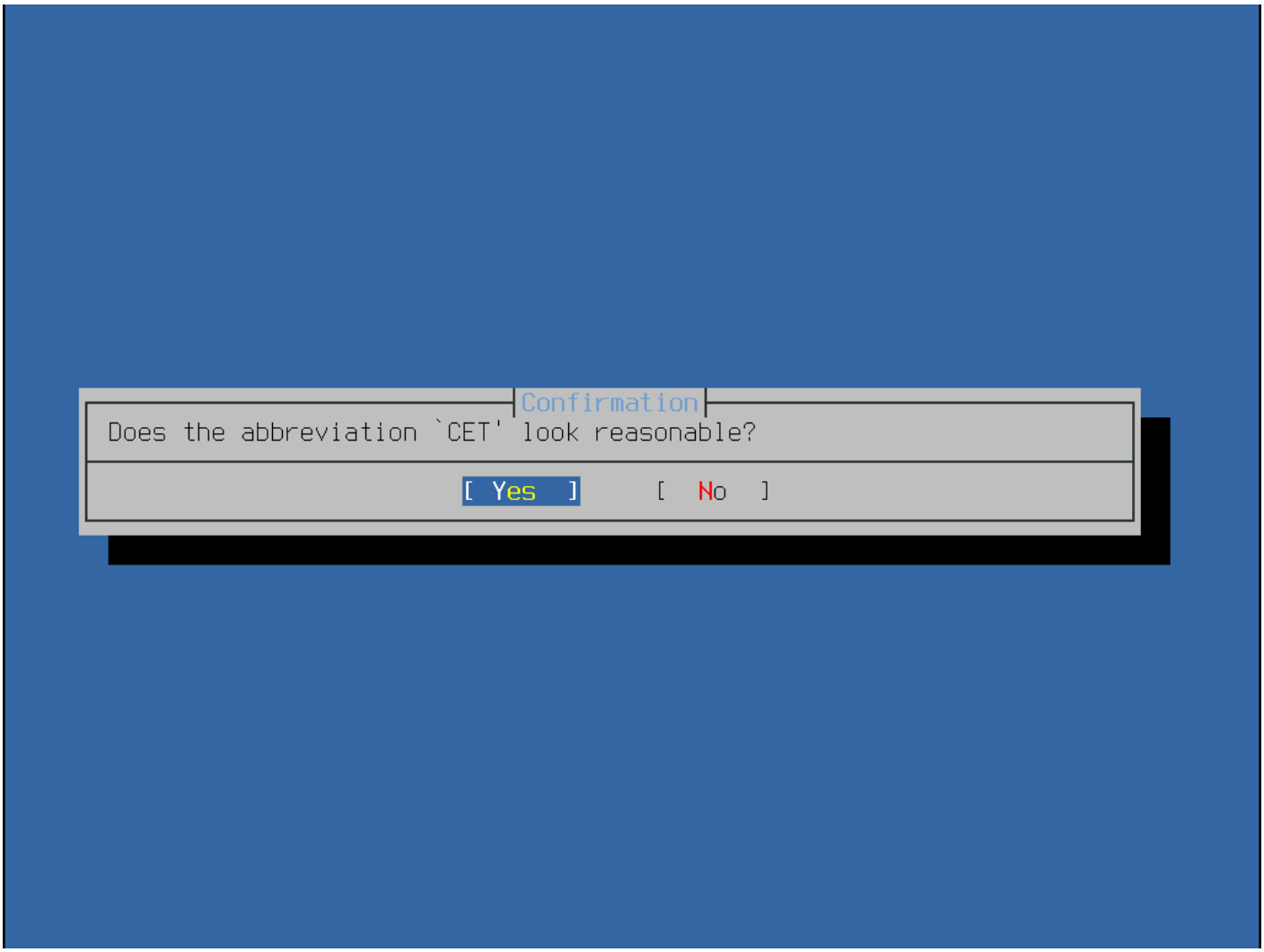


Figure 50. Confirm Time Zone

표준 시간대의 약어가 올바른지 확인합니다 .

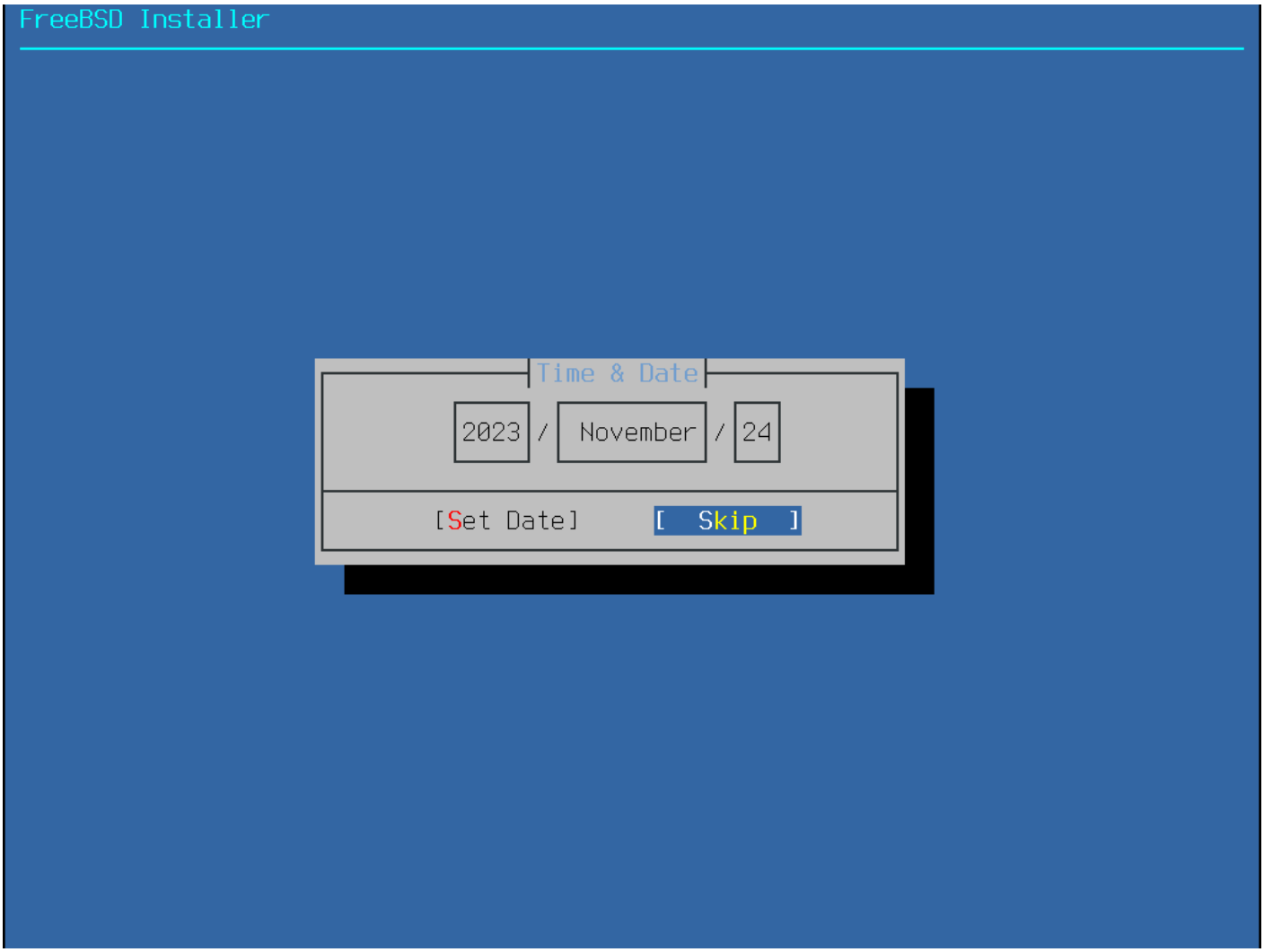


Figure 51. Select Date

화살표 키를 사용하여 적절한 날짜를 선택한 다음 **Set Date**을 누릅니다 . 그렇지 않으면 **Skip**을 눌러 날짜 선택을 건너뛸 수 있습니다 .

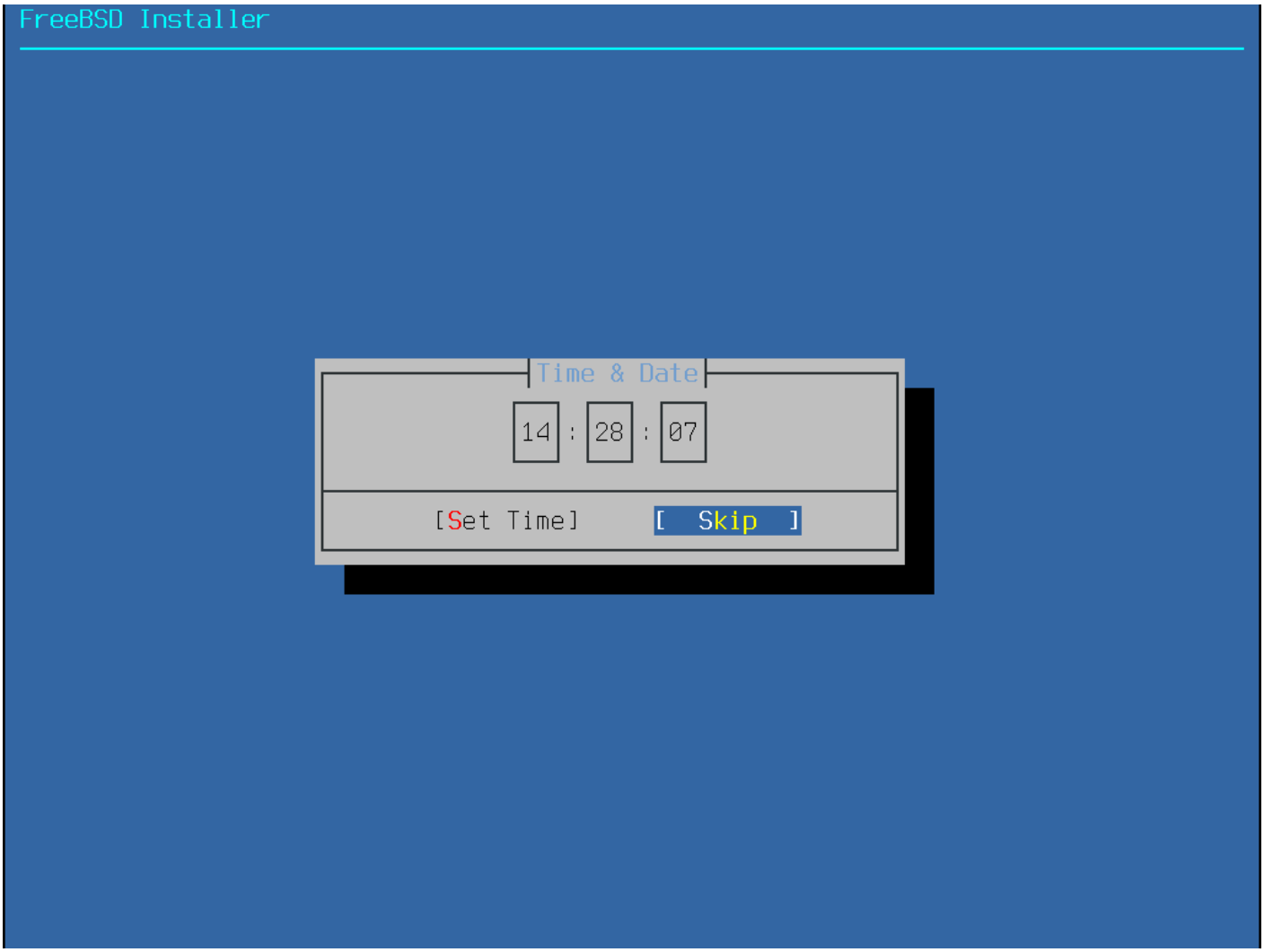


Figure 52. Select Time

화살표 키를 사용하여 적절한 시간을 선택한 다음 **Set Time**을 누릅니다 . 그렇지 않으면 **Skip**를 눌러 시간 선택을 건너뛸 수 있습니다 .

2.8.4. 서비스 활성화하기

다음 메뉴는 시스템이 부팅될 때마다 시작할 시스템 서비스를 구성하는 데 사용됩니다 . 이러한 서비스는 모두 선택 사항입니다 . 시스템 작동에 필요한 서비스만 시작하세요 .

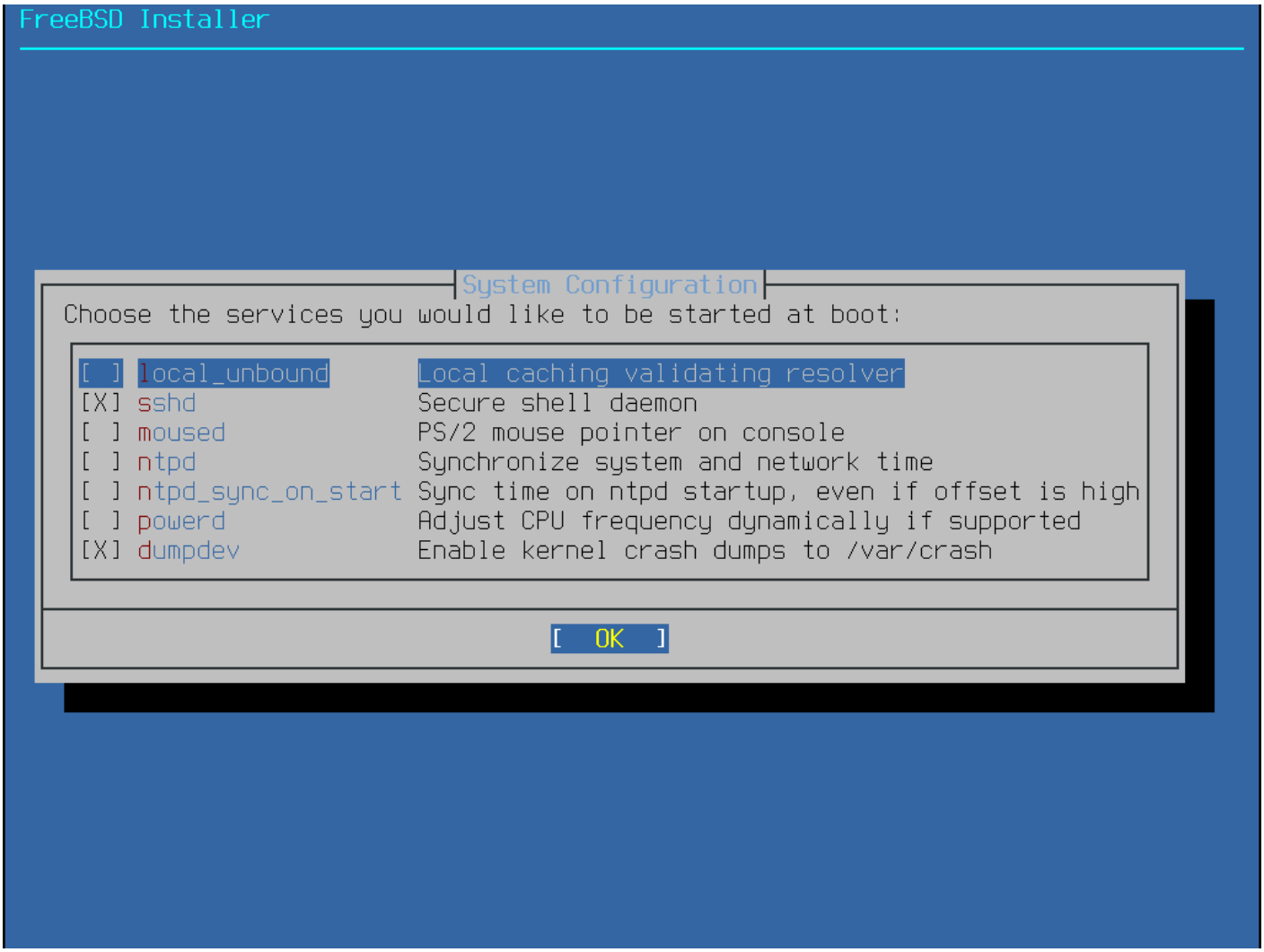


Figure 53. Selecting Additional Services to Enable

다음은 이 메뉴에서 사용 설정할 수 있는 서비스에 대한 요약입니다 :

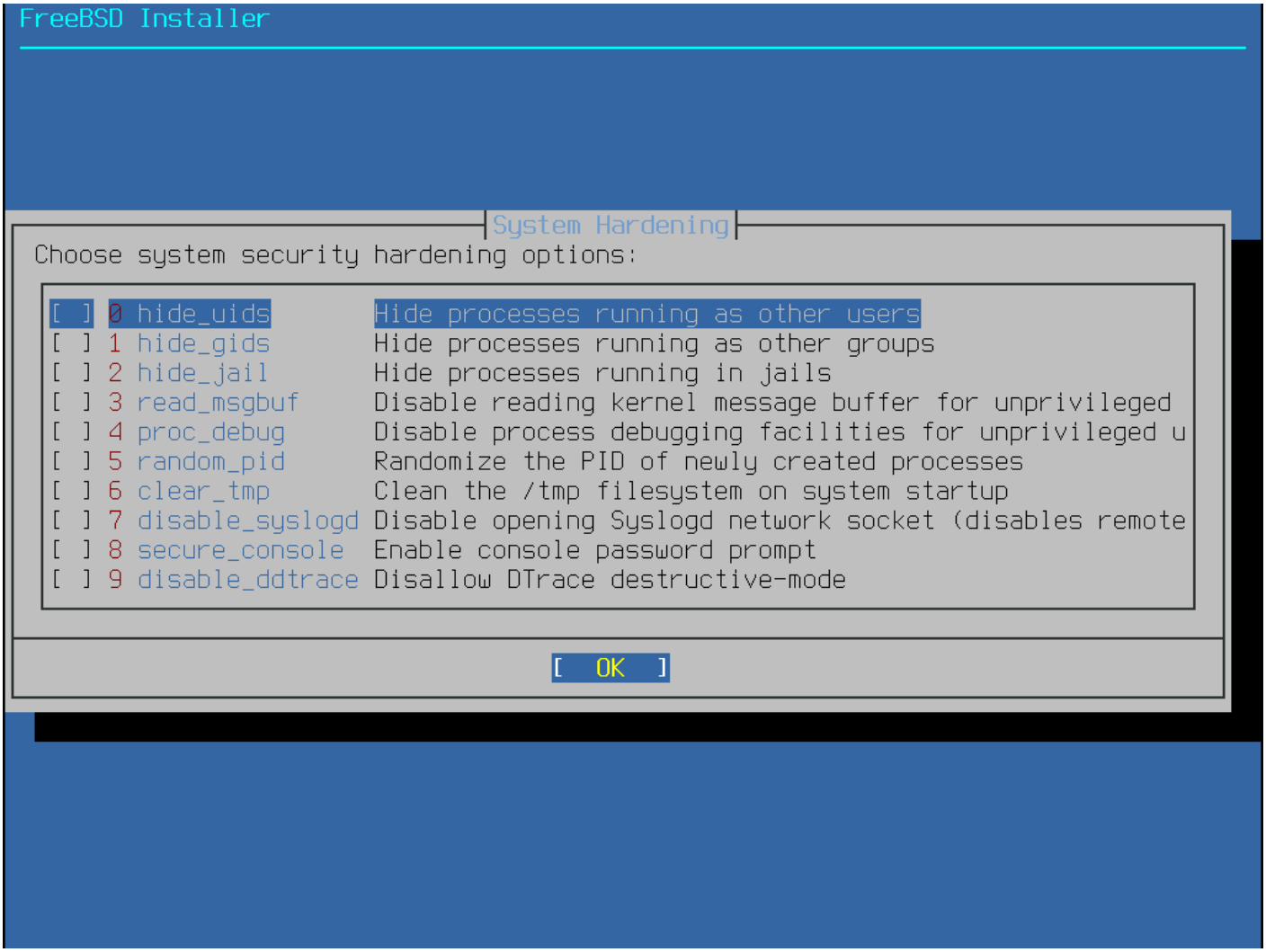
- local_unbound - DNS 로컬 언바인딩을 사용 설정합니다 . 이 설정은 로컬 캐싱 포워딩 확인자로만 사용하기 위한 구성이라는 점을 명심해야 합니다 . 전체 네트워크에 대한 확인자를 설정하는 것이 목적이라면 dns/unbound를 설치하세요 .

- sshd - 보안 셸 (SSH) 디먼은 암호화된 연결을 통해 시스템에 원격으로 액세스하는 데 사용됩니다 . 시스템이 원격 로그인에 사용 가능해야 하는 경우에만 이 서비스를 사용하도록 설정하세요 .

- moused - 명령줄 시스템 콘솔에서 마우스를 사용할 경우 이 서비스를 사용하도록 설정합니다 .

- ntpdate - 부팅 시 자동 시계 동기화를 사용하도록 설정합니다 . 이 프로그램의 기능은 이제 ntpd(8) 데몬에서 사용할 수 있으며 ntpdate(8) 유틸리티는 곧 폐기될 예정입니다 .

- ntpd - 자동 시계 동기화를 위한 네트워크 시간 프로토콜 (NTP) 데몬입니다 . 시스템 시계를 원격 시간 서버 또는 풀과 동기화하려면 이 서비스를 활성화하세요 .



- powerd - 전원 제어 및 에너지 절약을 위한 시스템 전원 제어 유틸리티입니다 .
- dumpdev - 크래시 덤프는 시스템 문제를 디버깅할 때 유용하므로 사용자가 활성화하는 것이 좋습니다 .

2.8.5. 보안 강화 옵션 활성화

다음 메뉴는 활성화할 보안 옵션을 구성하는 데 사용됩니다 . 이 모든 옵션은 선택 사항입니다 . 하지만 사용을 권장합니다 .

Figure 54. Selecting Hardening Security Options

다음은 이 메뉴에서 활성화할 수 있는 옵션의 요약입니다 :

- hide_uids - 다른 사용자 (UID)로 실행 중인 프로세스를 숨깁니다 . 이렇게 하면 권한이 없는 사용자가 다른 사용자의 실행 중인 프로세스를 볼 수 없습니다 .
- hide_gids - 다른 그룹 (GID)으로 실행 중인 프로세스를 숨깁니다 . 권한이 없는 사용자가 다른 그룹에서 실행 중인 프로세스를 볼 수 없도록 합니다 .

- hide_jail - jail에서 실행 중인 프로세스를 숨깁니다 . 권한이 없는 사용자가 감옥에서 실행 중인 프로세스를 볼 수 없도록 합니다 .
- read_msgbuf - 권한이 없는 사용자의 커널 메시지 버퍼 읽기를 비활성화합니다 . 권한이 없는 사용자가 dmesg(8)를 사용하여 커널의 로그 버퍼에서 메시지를 볼 수 없도록 합니다 .
- proc_debug - 권한이 없는 사용자의 프로세스 디버깅 기능을 비활성화합니다 . 일부 procfs 기능 , ptrace() 및 ktrace()를 포함한 다양한 권한 없는 프로세스 간 디버깅 서비스를 비활성화합니다 . 이 옵션을 사용하면 lldb(1), truss(1), procstat(1) 같은 디버깅 도구와 PHP 같은 특정 스크립팅 언어에 내장된 일부 디버깅 기능도 사용할 수 없게 된다는 점에 유의하세요 .
- random_pid - 프로세스의 PID를 무작위화합니다 .
- clear_tmp - 시스템 시작 시 /tmp를 정리합니다 .
- disable_syslogd - syslogd 네트워크 소켓 열기를 비활성화합니다 . 기본적으로 FreeBSD는 -s를 사용하여 안전한 방식으로 syslogd를 실행합니다 . 이렇게 하면 데몬이 포트 514에서 들어오는 UDP 요청을 수신 대기하지 않습니다 . 이 옵션을 활성화하면 -ss를 사용하여 syslogd가 대신 실행되며 , 이는 syslogd가 포트를 열지 못하도록 합니다 . 자세한 내용은 syslogd(8)을 참조하세요 .
- disable_sendmail - sendmail 메일 전송 에이전트를 비활성화합니다 .
- secure_console - 단일 사용자 모드로 들어갈 때 명령 프롬프트에서 root 비밀번호를 요청하도록 합니다 .
- disable_ddtrace - 실행 중인 커널에 영향을 미치는 모드에서 DTrace를 실행할 수 있습니다 . 명시적으로 활성화하지 않으면 파괴적인 작업을 사용할 수 없습니다 . DTrace를 사용할 때 이 옵션을 활성화하려면 -w를 사용하세요 . 자세한 내용은 dtrace(1)을 참조하세요 .
- enable_aslr - 주소 레이아웃 무작위화를 활성화합니다 . 주소 레이아웃 무작위화에 대한 자세한 내용은 위키백과 문서 를 참조하세요 .

2.8.6. 사용자 추가

다음 메뉴에서 하나 이상의 사용자 계정을 만들라는 메시지가 표시됩니다 . root가 아닌 사용자 계정을 사용하여 시스템에 로그인하는 것이 좋습니다 . root로 로그인하면 기본적으로 수행할 수 있는 작업에 대한 제한이나 보호가 없습니다 . 일반 사용자로 로그인하는 것이 더 안전하고 안전합니다 .

새 사용자를 추가하려면 **Yes**를 선택합니다 .

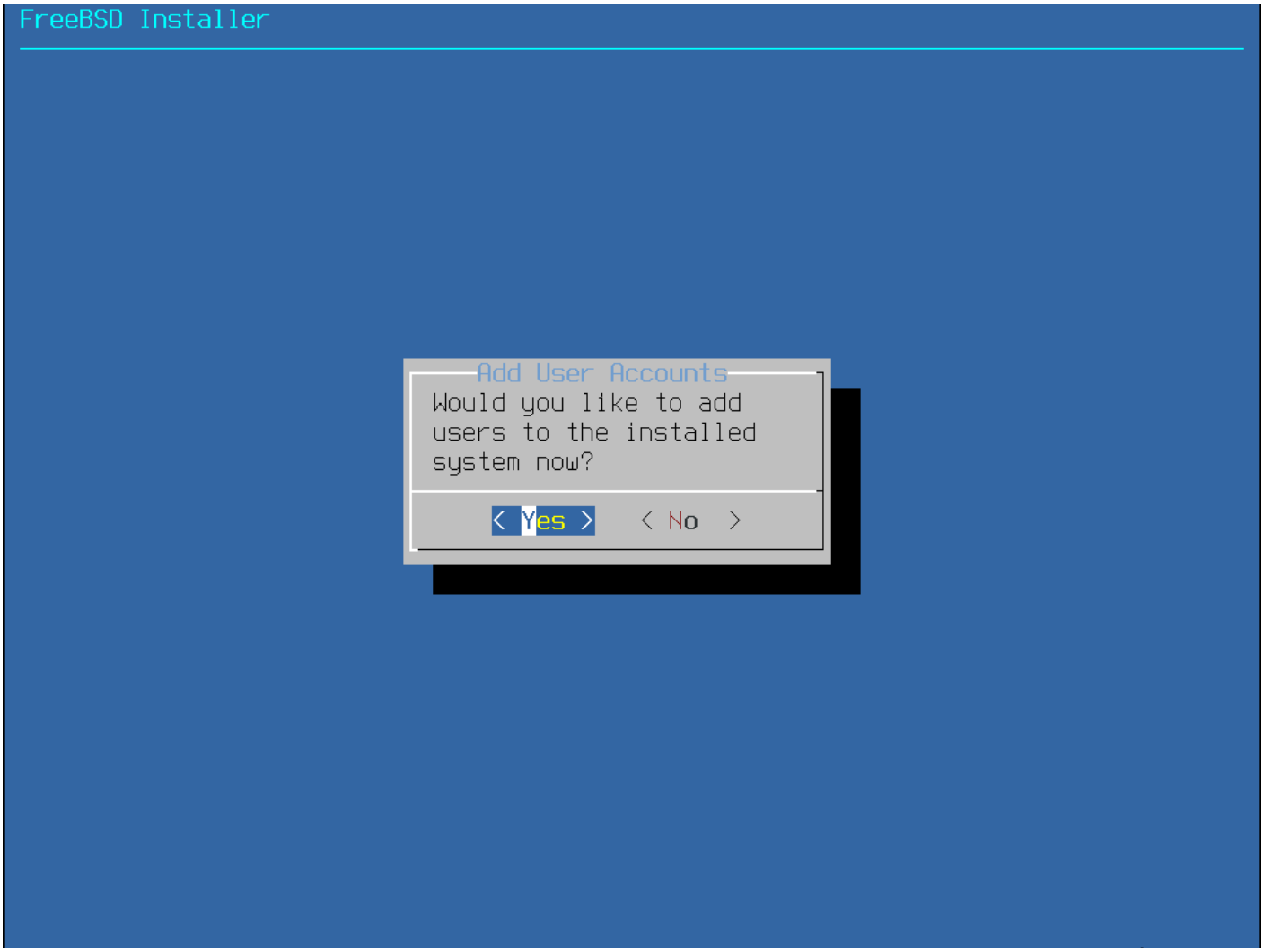


Figure 55. Add User Accounts

메시지에 따라 사용자 계정에 대해 요청된 정보를 입력합니다 . 사용자 정보 입력에 표시된 예는 예제 사용자 계정을 만듭니다 .

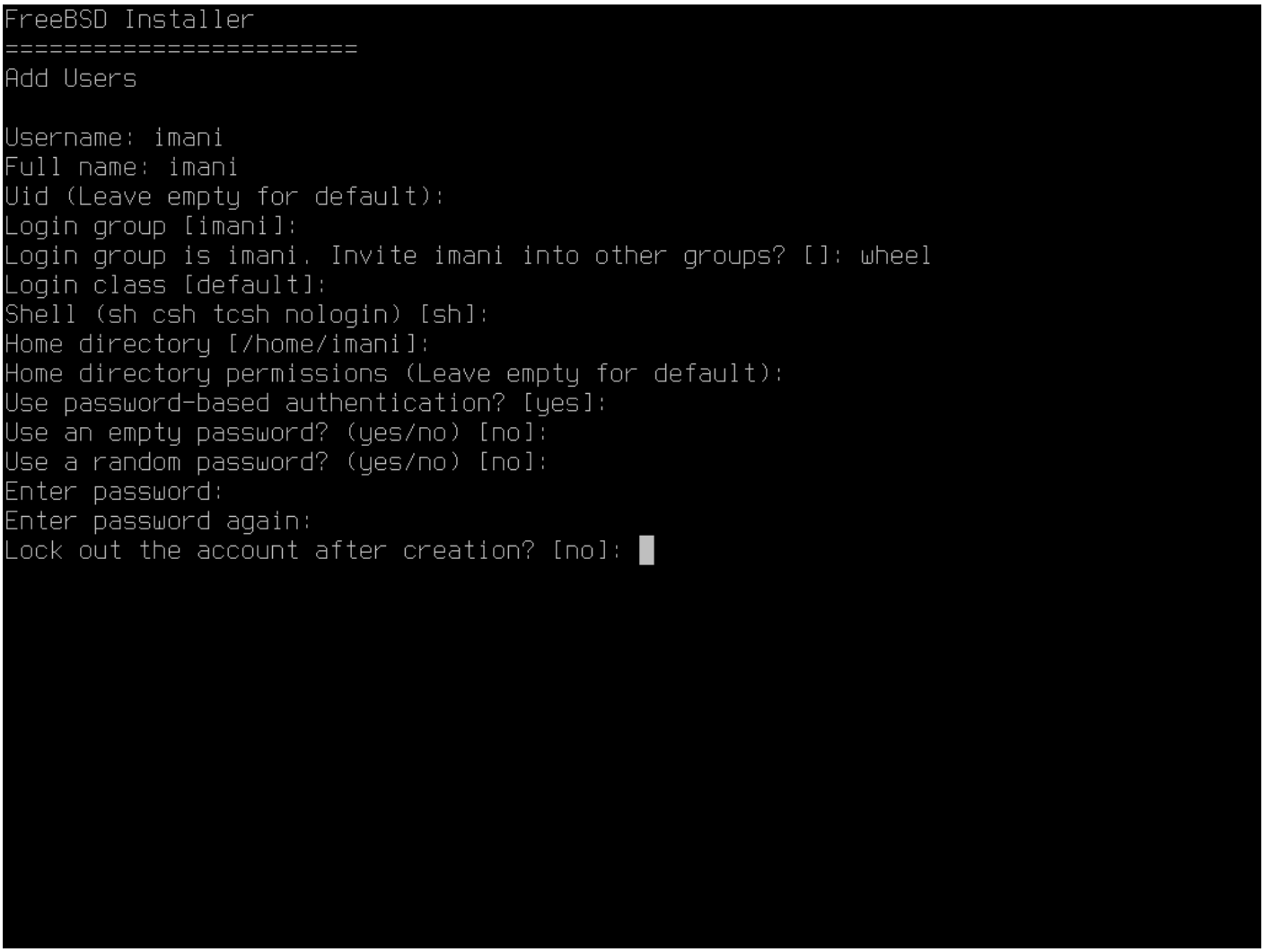


Figure 56. Enter User Information

다음은 입력할 정보를 요약한 것입니다 :

- **Username** - 사용자가 로그인할 때 입력하는 이름입니다 . 일반적인 규칙은 각 사용자 아이디가 시스템에서 고유한 이름인 한 이름의 첫 글자와 성을 결합하여 사용하는 것입니다 . 사용자 아이디는 대소문자를 구분하며 공백을 포함하지 않아야 합니다 .
- **Full name** - 사용자의 전체 이름입니다 . 공백을 포함할 수 있으며 사용자 계정에 대한 설명으로 사용됩니다 .
- **Uid** - 사용자 ID입니다 . 일반적으로 비워두면 시스템에서 자동으로 값을 할당합니다 .
- **Login group** - 사용자의 그룹입니다 . 일반적으로 기본값을 적용하기 위해 비워 두는 것이 일반적입니다 .
- **Invite user into other groups?** - 사용자를 회원으로 추가할 추가 그룹입니다 . 사용자에게 관리자 액세스 권한이 필요한 경우 여기에 wheel을 입력합니다 .
- **Login class** - 일반적으로 기본값으로 비워 둡니다 .

- **Shell** - [illegible] [illegible] [illegible]

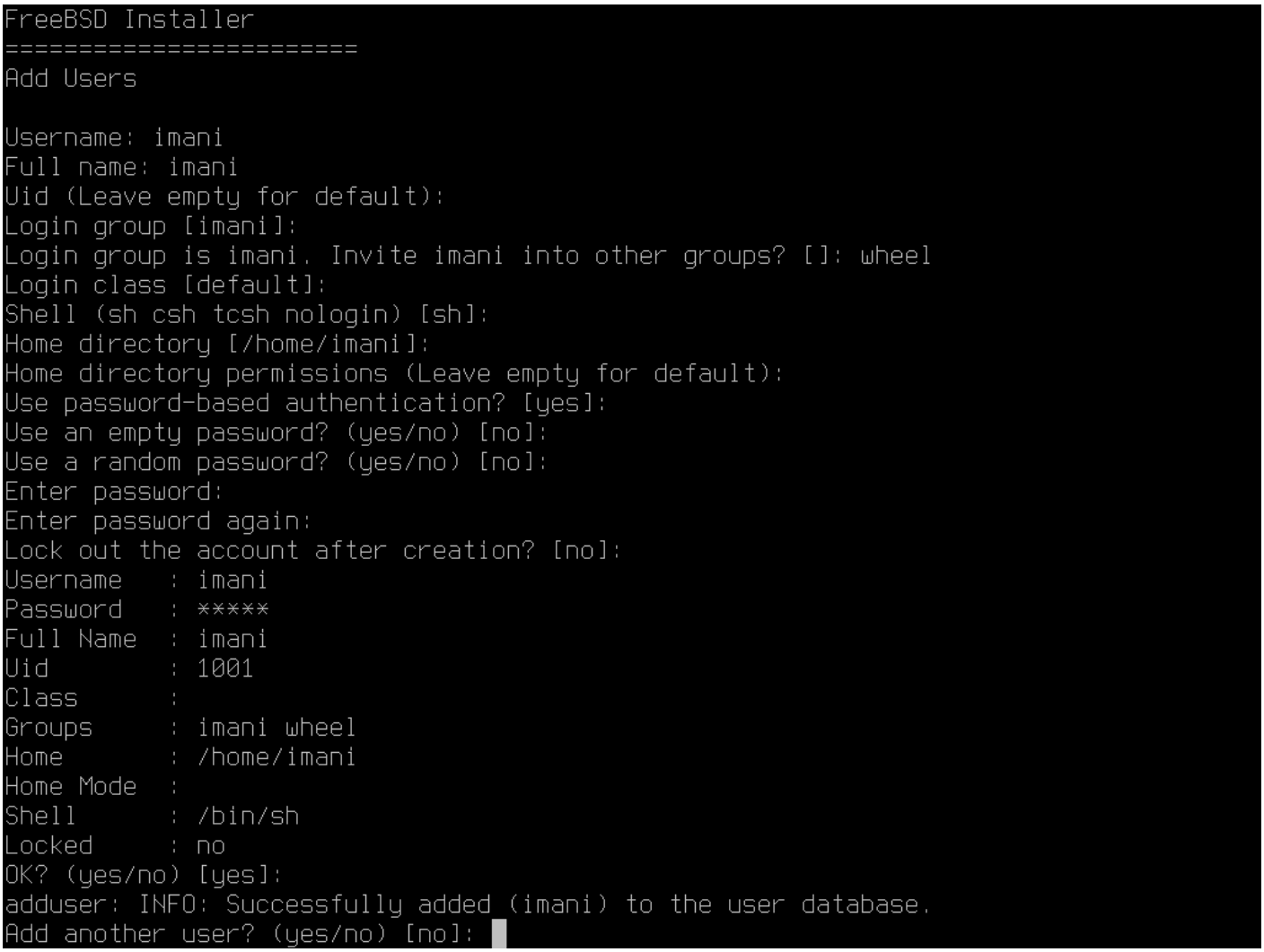


Figure 57. Exit User and Group Management

추가할 사용자가 더 있으면 **Add another user?** 질문에 Yes로 대답합니다 . No를 입력하면 사용자 추가를 완료하고 설치를 계속 진행합니다 .

사용자 추가 및 사용자 관리에 대한 자세한 내용은 사용자 및 기본 계정 관리를 참조하세요 .

2.8.7. 최종 구성

모든 설치 및 구성이 완료되면 마지막으로 설정을 수정할 수 있는 기회가 제공됩니다 .

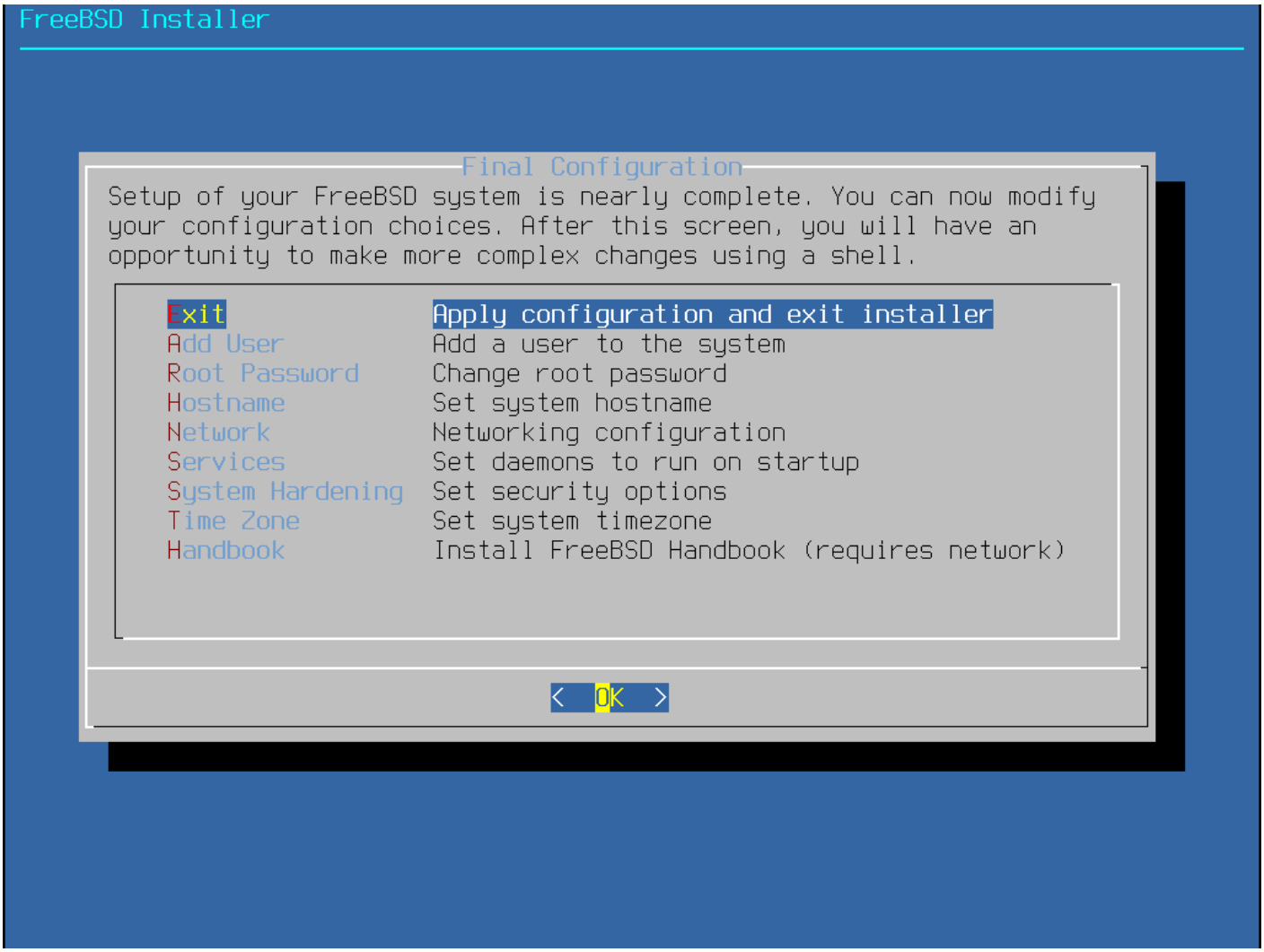


Figure 58. Final Configuration

이 메뉴를 사용하여 설치를 완료하기 전에 변경하거나 추가 구성을 수행합니다 .

- **Add User** - 사용자 추가에 설명되어 있습니다 .
- **Root Password** - 루트 비밀번호 설정에 설명되어 있습니다 .
- **Hostname**- 호스트 이름 설정에 설명되어 있습니다 .
- **Network** - 네트워크 인터페이스 구성에 설명되어 있습니다 .
- **Services** - 서비스 활성화에 설명되어 있습니다 .
- **System Hardening** - 보안 옵션 강화 활성화에 설명되어 있습니다 .
- **Time Zone** - 표준 시간대 설정에 설명되어 있습니다 .
- **Handbook** - FreeBSD 핸드북을 다운로드하여 설치합니다 .

구성이 완료되면 Exit를 선택합니다 .

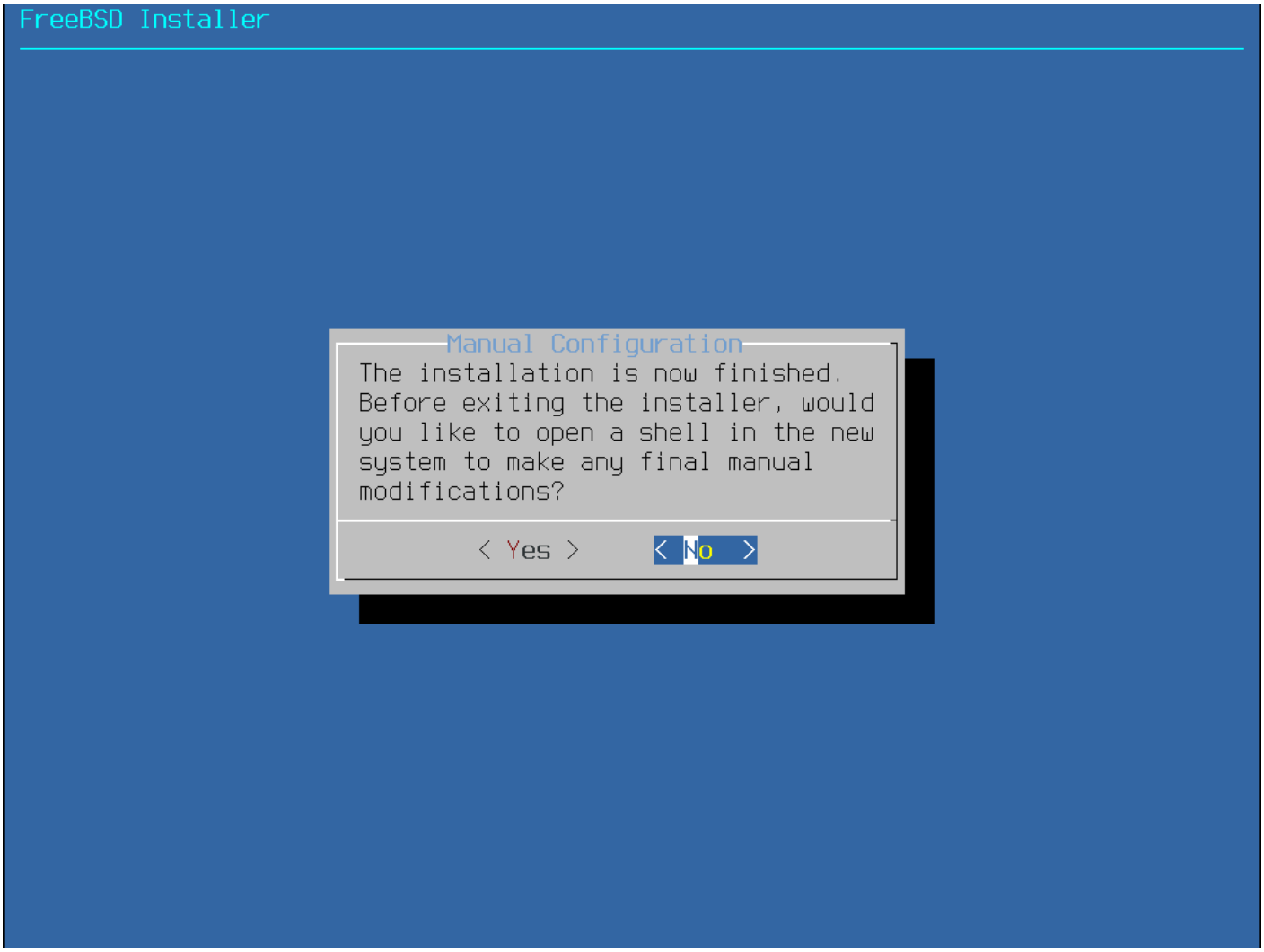


Figure 59. Manual Configuration

새 시스템으로 재부팅하기 전에 수행해야 할 추가 구성이 있는지 묻는 메시지가 표시됩니다 . 새 시스템 내의 셸로 종료하려면 **Yes**를 선택하고 , 설치의 마지막 단계로 진행하려면 **No**를 선택합니다 .

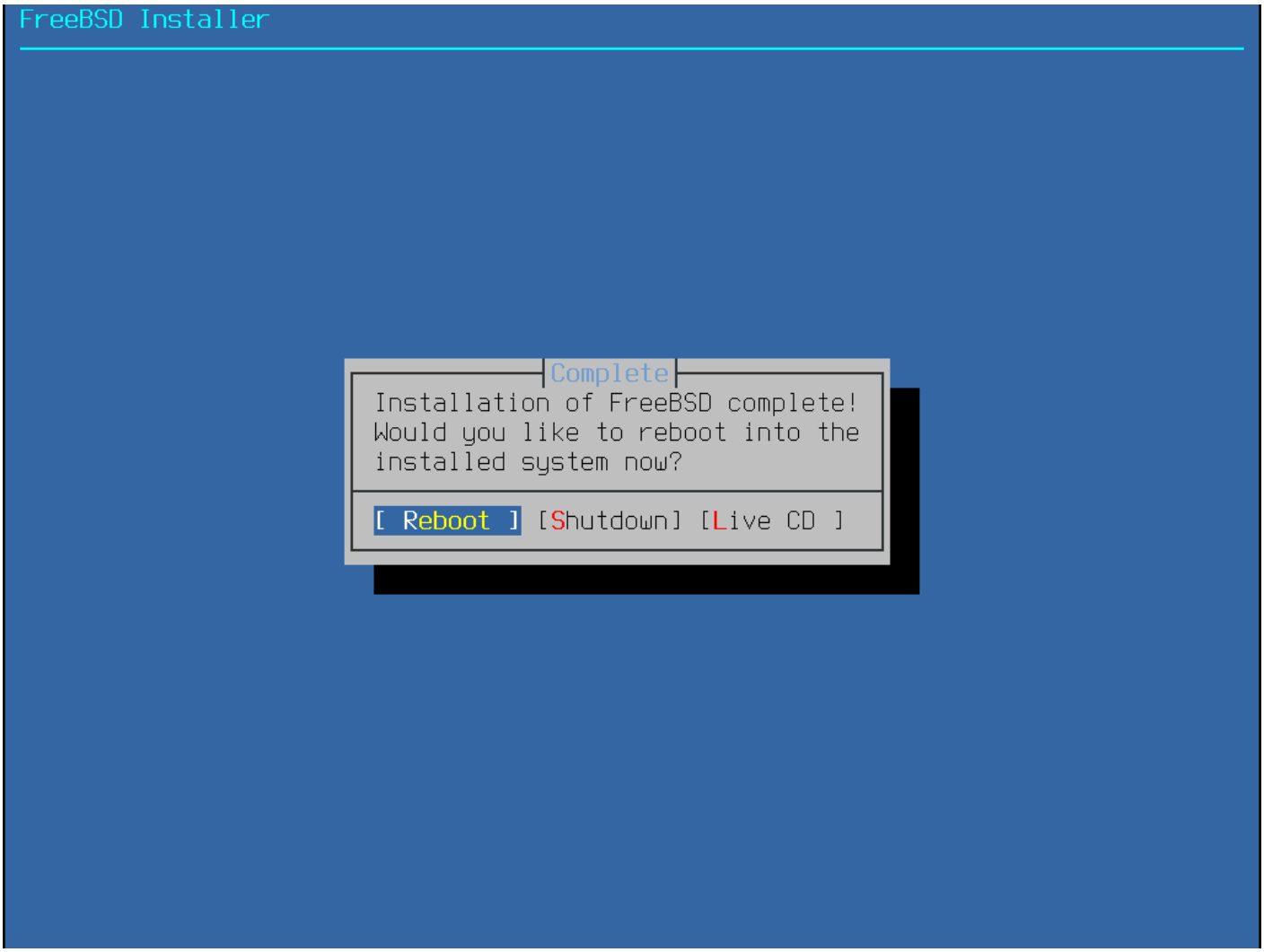


Figure 60. Complete the Installation

추가 구성이나 특별한 설정이 필요한 경우 **Live CD**를 선택하여 설치 미디어를 라이브 CD 모드로 부팅합니다 .

설치가 완료되면 **Reboot**을 선택하여 컴퓨터를 재부팅하고 새 FreeBSD 시스템을 시작합니다 . FreeBSD 설치 미디어를 제거하지 않으면 컴퓨터가 다시 부팅될 수 있으니 잊지 마세요 .

FreeBSD가 부팅되면 정보 메시지가 표시됩니다 . 시스템 부팅이 완료되면 **login:** 프롬프트가 표시됩니다 . **login:** 프롬프트에서 설치 중에 추가한 사용자 이름을 입력합니다 . **root**로 로그인하지 마세요 . 관리 액세스 권한이 필요한 경우 수퍼유저가 되는 방법에 대한 지침은 수퍼유저 계정을 참조하세요 .

부팅하는 동안 표시되는 메시지는 `Scroll-Lock`을 눌러 스크롤백 버퍼를 켜서 검토할 수 있습니다 . PgUp, PgDn 및 화살표 키를 사용하여 메시지를 뒤로 스크롤할 수 있습니다 . 완료하면 `Scroll-Lock`을 다시 눌러 디스플레이 잠금을 해제하고 콘솔로 돌아갑니다 . 시스템이 잠시 가동된 후 이러한 메시지를 검토하려면 명령 프롬프트에서 `less /var/run/dmesg.boot`를 입력합니다 . 확인 후 명령줄로 돌아가려면 **q**를 누릅니다 .

활성화할 추가 서비스 선택에서 sshd를 활성화한 경우 시스템이 SSH 호스트 키를 생성하는 동안 첫 번째 부팅이 약간 느려질 수 있습니다 . 이후 부팅은 더 빨라집니다 . 그러면 다음 예와 같이 키의 지문이 표시됩니다 :

```
Generating public/private rsa1 key pair.
Your identification has been saved in /etc/ssh/ssh_host_key.
```

```
Your public key has been saved in /etc/ssh/ssh_host_key.pub.
The key fingerprint is:
10:a0:f5:af:93:ae:a3:1a:b2:bb:3c:35:d9:5a:b3:f3 root@machine3.example.com
The key's randomart image is:
+--[RSA1 1024]----+
|    o..          |
|   o . .         |
|  .   o          |
|      o          |
|    o   S        |
|   + + o         |
|o . + *          |
|o+ ..+ .         |
|==o..o+E         |
+-----------------+
Generating public/private dsa key pair.
Your identification has been saved in /etc/ssh/ssh_host_dsa_key.
Your public key has been saved in /etc/ssh/ssh_host_dsa_key.pub.
The key fingerprint is:
7e:1c:ce:dc:8a:3a:18:13:5b:34:b5:cf:d9:d1:47:b2 root@machine3.example.com
The key's randomart image is:
+--[ DSA 1024]----+
|      ..     . .|
|     o  .   . + |
|    . ..   . E .|
|   . .  o o . . |
|    +  S = .    |
|   +  . = o     |
|    +  . * .    |
|   . .  o .     |
|     .o. .      |
+-----------------+
Starting sshd.
```

지문 및 SSH에 대한 자세한 내용은 OpenSSH를 참조하세요 .

FreeBSD는 기본적으로 그래픽 환경을 설치하지 않습니다 . 그래픽 창 관리자의 설치 및 구성에 대한 자세한 내용은 X 윈도우 시스템을 참조하십시오 .

FreeBSD 컴퓨터를 올바르게 종료하면 데이터와 하드웨어가 손상되지 않도록 보호할 수 있습니다 . 시스템이 제대로 종료되기 전에 전원을 끄지 마세요 ! 사용자가 wheel 그룹의 구성원인 경우 , 명령줄에 su를 입력하고 루트 비밀번호를 입력하여 수퍼유저가 됩니다 . 그런 다음 shutdown -p now를 입력하면 시스템이 깨끗하게 종료되고 하드웨어가 이를 지원하는 경우 자동으로 꺼집니다 .

2.9. 문제 해결

이 섹션에서는 사람들이 보고한 일반적인 문제와 같은 기본적인 설치 문제 해결 방법을 다룹니다 .

하드웨어가 지원되는지 확인하려면 FreeBSD 릴리스 정보 페이지에 나와 있는 하드웨어 노트에서 해당 버전의 FreeBSD를 확인하세요 .

Note

다양한 하드웨어 구성 요소 , 특히 마더보드의 펌웨어를 업데이트하면 일부 설치 문제를 피하거나 완화할 수 있습니다 . 마더보드 펌웨어는 일반적으로 BIOS라고 합니다 . 대부분의 마더보드 및 컴퓨터 제조업체는 업그레이드 및 업그레이드 정보를 제공하는 웹사이트를 운영하고 있습니다 .

제조업체는 일반적으로 중요한 업데이트와 같은 정당한 사유가 없는 한 마더보드 BIOS를 업그레이드하지 말 것을 권장합니다 . 업그레이드 프로세스가 잘못 진행되어 BIOS가 불완전해지고 컴퓨터가 작동하지 않을 수 있습니다 .

부팅 중 하드웨어를 검사하는 동안 시스템이 멈추거나 설치 과정에서 이상하게 동작하는 경우 , ACPI가 원인일 수 있습니다 . FreeBSD는 i386 및 amd64 플랫폼에서 시스템 ACPI 서비스를 광범위하게 사용하여 부팅 중에 시스템 구성이 감지될 경우 이를 지원합니다 . 안타깝게도 일부 버그는 여전히 ACPI 드라이버와 시스템 마더보드 및 BIOS 펌웨어에 존재합니다 . ACPI는 3단계 부트로더에서 **hint.acpi.0.disabled** 힌트를 설정하여 비활성화할 수 있습니다 :

```
set hint.acpi.0.disabled="1"
```

2.10. 라이브 CD 사용하기

시작 메뉴에 표시된 bsdinstall의 시작 메뉴에는 라이브 CD 옵션이 있습니다 . 이 옵션은 FreeBSD가 자신에게 적합한 운영체제인지 궁금하고 설치하기 전에 몇 가지 기능을 테스트해보고 싶은 분들에게 유용합니다 .

라이브 CD를 사용하기 전에 다음 사항에 유의해야 합니다 :

- 시스템에 액세스하려면 인증이 필요합니다 . 사용자 아이디는 루트이고 비밀번호는 공백입니다 .
- 시스템이 설치 미디어에서 직접 실행되므로 하드 디스크에 설치된 시스템보다 성능이 상당히 느려집니다 .

- 이 옵션은 그래픽 인터페이스가 아닌 명령 프롬프트만 제공합니다 .

Last modified on: January 6, 2024 by Danilo G. Baio

Chapter 3. FreeBSD 기초

3.1. 요약

이 장에서는 FreeBSD 운영 체제의 기본 명령어와 기능에 대해 설명합니다. 이 자료의 대부분은 모든 UNIX® 계열 운영 체제와 관련이 있습니다. FreeBSD를 처음 사용하는 사용자는 이 장을 주의 깊게 읽어 보시기 바랍니다.

이 장을 읽고 나면 다음과 같은 내용을 알게 될 것입니다:

- 가상 콘솔을 사용하고 구성하는 방법.
- FreeBSD에서 사용자 및 그룹을 생성하고 관리하는 방법.
- UNIX® 파일 권한 및 FreeBSD 파일 플래그의 작동 방식.
- 기본 FreeBSD 파일 시스템 레이아웃.
- FreeBSD 디스크 구성.
- 파일 시스템을 마운트 및 마운트 해제하는 방법.
- 프로세스, 데몬 및 신호가 무엇인지.
- 셸이란 무엇이며 기본 로그인 환경을 변경하는 방법.
- 기본 텍스트 편집기를 사용하는 방법.
- 장치와 장치 노드란 무엇인가.
- 매뉴얼 페이지를 참조하는 방법.

3.2. 가상 콘솔과 터미널

시작 시 그래픽 환경을 자동으로 시작하도록 FreeBSD를 구성하지 않은 경우, 이 예제에서 볼 수 있듯이 시스템이 명령줄 로그인 프롬프트로 부팅됩니다:

```
FreeBSD/amd64 (pc3.example.org) (ttyv0)

login:
```

첫 번째 줄에는 시스템에 대한 몇 가지 정보가 포함되어 있습니다. amd64는 FreeBSD가 64비트 x86 시스템에서 실행 중임을 나타냅니다. 호스트 이름은 pc3.example.org이고, ttyv0은 이것이 "시스템 콘솔" 임을 나타냅니다. 두 번째 줄은 로그인 프롬프트입니다.

FreeBSD는 다중 사용자 시스템이기 때문에 다른 사용자를 구별할 수 있는 방법이 필요합니다. 이는 시스템의 프로그램에 액세스하기 전에 모든 사용자가 시스템에 로그인하도록 요구함으로써 이루어집니다. 모든 사용자에게는 고유한 '사용자 아이디'와 개인 '비밀번호'가 있습니다.

시스템 콘솔에 로그인하려면 사용자 추가에 설명된 대로 시스템 설치 시 구성한 사용자 아이디를 입력하고 Enter 키를 누릅니다. 그런 다음 사용자 아이디와 연결된 비밀번호를 입력하고 **Enter** 키를 누릅니다. 비밀번호는 보안상의 이유로 에코되지 않습니다.

올바른 비밀번호를 입력하면 오늘의 메시지 (MOTD)가 표시되고 명령 프롬프트가 이어집니다. 사용자 생성 시 선택한 셸에 따라 이 프롬프트는 #, $ 또는 % 문자가 됩니다. 이 프롬프트는 사용자가 이제 FreeBSD 시스템 콘솔에 로그인하여 사용 가능한 명령을 시도할 준비가 되었음을 나타냅니다.

3.2.1. 가상 콘솔

시스템 콘솔을 사용하여 시스템과 상호작용할 수 있지만, FreeBSD 시스템의 키보드에서 명령줄로 작업하는 사용자는 일반적으로 가상 콘솔에 로그인합니다. 이는 시스템 메시지가 기본적으로 시스템 콘솔에 표시되도록 구성되어 있기 때문입니다. 이러한 메시지는 사용자가 작업 중인 명령이나 파일 위에 표시되므로 당면한 작업에 집중하기 어렵습니다.

기본적으로 FreeBSD는 명령 입력을 위한 여러 가상 콘솔을 제공하도록 구성됩니다. 각 가상 콘솔에는 고유한 로그인 프롬프트와 셸이 있으며 가상 콘솔 간에 쉽게 전환할 수 있습니다. 이는 기본적으로 그래픽 환경에서 여러 개의 창을 동시에 열어 놓은 것과 같은 명령줄을 제공합니다.

Alt+F1 ~ Alt+F8 키 조합은 가상 콘솔 간 전환을 위해 FreeBSD에 예약되어 있습니다. 시스템 콘솔 (ttyv0)로 전환하려면 Alt+F1, 첫 번째 가상 콘솔 (ttyv1)에 액세스하려면 Alt+F2, 두 번째 가상 콘솔 (ttyv2)에 액세스하려면 Alt+F3 등을 사용하세요. Xorg를 그래픽 콘솔로 사용하는 경우, 텍스트 기반 가상 콘솔로 돌아가려면 Ctrl+Alt+F1을 조합하면 됩니다.

한 콘솔에서 다른 콘솔로 전환할 때 FreeBSD가 화면 출력을 관리합니다. 그 결과 여러 개의 가상 화면과 키보드가 있는 것처럼 착각하여 명령을 입력할 수 있는 FreeBSD를 실행할 수 있습니다. 하나의 가상 콘솔에서 실행된 프로그램은 사용자가 다른 가상 콘솔로 전환해도 실행이 중지되지 않습니다.

FreeBSD 콘솔과 키보드 드라이버에 대한 자세한 기술 설명은 kbdcontrol(1), vidcontrol(1), atkbd(4), syscons(4) 및 vt(4)를 참고하세요.

FreeBSD에서 사용 가능한 가상 콘솔의 수는 /etc/ttys의 이 섹션에서 구성할 수 있습니다:

```
# name    getty                    type  status comments
#
ttyv0   "/usr/libexec/getty Pc"        xterm   on  secure
# Virtual terminals
ttyv1   "/usr/libexec/getty Pc"        xterm   on  secure
ttyv2   "/usr/libexec/getty Pc"        xterm   on  secure
ttyv3   "/usr/libexec/getty Pc"        xterm   on  secure
ttyv4   "/usr/libexec/getty Pc"        xterm   on  secure
ttyv5   "/usr/libexec/getty Pc"        xterm   on  secure
ttyv6   "/usr/libexec/getty Pc"        xterm   on  secure
ttyv7   "/usr/libexec/getty Pc"        xterm   on  secure
ttyv8   "/usr/X11R6/bin/xdm -nodaemon"  xterm   off secure
```

가상 콘솔을 비활성화하려면 해당 가상 콘솔을 나타내는 줄의 시작 부분에 주석 기호 (#)를 넣습니다 . 예를 들어 사용 가능한 가상 콘솔 수를 8개에서 4개로 줄이려면 가상 콘솔 ttyv5부터 ttyv8까지를 나타내는 마지막 네 줄 앞에 #을 넣습니다 . 시스템 콘솔 ttyv0에 대한 줄은 주석 처리하지 마십시오 . Xorg가 X Window 시스템에 설명된 대로 설치 및 구성된 경우 마지막 가상 콘솔 (ttyv8)은 그래픽 환경에 액세스하는 데 사용됩니다 .

이 파일의 모든 열에 대한 자세한 설명과 가상 콘솔에 사용할 수 있는 옵션은 ttys(5)를 참조하세요 .

3.2.2. 단일 사용자 모드

FreeBSD 부팅 메뉴에는 "Boot Single User"이라고 표시된 옵션이 있습니다 . 이 옵션을 선택하면 시스템이 "Boot Single User"라는 특수 모드로 부팅됩니다 . 이 모드는 일반적으로 부팅되지 않는 시스템을 복구하거나 root 비밀번호를 알 수 없는 경우 재설정하는 데 사용됩니다 . 단일 사용자 모드에서는 네트워킹 및 기타 가상 콘솔을 사용할 수 없습니다 . 그러나 시스템에 대한 전체 루트 액세스 권한은 사용할 수 있으며 기본적으로 root 비밀번호는 필요하지 않습니다 . 이러한 이유로 이 모드로 부팅하려면 키보드에 물리적으로 액세스해야 하며 , 키보드에 물리적으로 액세스할 수 있는 사람을 결정하는 것은 FreeBSD 시스템을 보호할 때 고려해야 할 사항입니다 .

단일 사용자 모드를 제어하는 설정은 /etc/ttys의 이 섹션에서 찾을 수 있습니다 :

```
# name  getty                    type  status  comments
#
# If console is marked "insecure", then init will ask for the root password
# when going to single-user mode.
console none                    unknown  off  secure
```

기본적으로 상태는 secure로 설정되어 있습니다 . 이는 키보드에 물리적으로 액세스할 수 있는 사람이 중요하지 않거나 물리적 보안 정책에 의해 제어된다고 가정합니다 . 이 설정을 안전하지 않음으로 변경하면 , 누구나 키보드에 액세스할 수 있기 때문에 환경 자체가 안전하지 않다고 가정합니다 . 이 줄을 insecure로

변경하면 사용자가 단일 사용자 모드로 부팅하도록 선택할 때 root 암호를 묻는 메시지가 표시됩니다 .

Note

이 설정을 *insecure*로 변경할 때는 주의하세요 ! 루트 비밀번호를 잊어버린 경우에도 단일 사용자 모드로 부팅할 수 있지만 , FreeBSD 부팅 프로세스에 익숙하지 않은 사람에게는 어려울 수 있습니다 .

3.2.3. 콘솔 비디오 모드 변경하기

FreeBSD 콘솔 기본 비디오 모드는 1024x768, 1280x1024 또는 그래픽 칩과 모니터에서 지원하는 다른 크기로 조정할 수 있습니다 . 다른 비디오 모드를 사용하려면 VESA 모듈을 로드하세요 :

```
# kldload vesa
```

하드웨어에서 지원되는 동영상 모드를 확인하려면 vidcontrol(1)을 사용합니다 . 지원되는 비디오 모드 목록을 보려면 다음을 실행하세요 :

```
# vidcontrol -i mode
```

이 명령의 출력에는 하드웨어에서 지원하는 비디오 모드가 나열됩니다 . 새 비디오 모드를 선택하려면 루트 사용자로 vidcontrol(1)을 사용하여 모드를 지정합니다 :

```
# vidcontrol MODE_279
```

새 비디오 모드가 적합하다면 /etc/rc.conf에 추가하여 부팅 시 영구적으로 설정할 수 있습니다 :

```
allscreens_flags="MODE_279"
```

3.3. 사용자 및 기본 계정 관리

FreeBSD는 여러 사용자가 동시에 컴퓨터를 사용할 수 있게 해줍니다 . 한 번에 한 명의 사용자만 화면 앞에 앉아 키보드를 사용할 수 있지만 , 네트워크를 통해 원하는 수의 사용자가 시스템에 로그인할 수 있습니다 . 시스템을 사용하려면 각 사용자에게 고유한 사용자 계정이 있어야 합니다 .

이 장에서는 이에 대해 설명합니다 :

- FreeBSD 시스템의 다양한 사용자 계정 유형 .
- 사용자 계정을 추가 , 제거 및 수정하는 방법 .

- 사용자 및 그룹이 액세스할 수 있는 리소스를 제어하기 위해 제한을 설정하는 방법 .
- 그룹을 만들고 사용자를 그룹의 구성원으로 추가하는 방법 .

3.3.1. 계정 유형

FreeBSD 시스템에 대한 모든 액세스는 계정을 사용하여 이루어지고 모든 프로세스는 사용자가 실행하기 때문에 사용자 및 계정 관리가 중요합니다 .

계정에는 시스템 계정 , 사용자 계정 , 수퍼유저 계정의 세 가지 주요 유형이 있습니다 .

3.3.1.1. 시스템 계정

시스템 계정은 DNS, 메일 , 웹 서버 등의 서비스를 실행하는 데 사용됩니다 . 모든 서비스가 수퍼유저로 실행되면 제한 없이 모든 작업을 수행할 수 있으므로 보안을 위해서입니다 .

시스템 계정의 예로는 daemon, operator, bind, news, www 등이 있습니다 .

nobody는 일반적인 권한 없는 시스템 계정입니다 . 그러나 nobody 계정을 사용하는 서비스가 많을수록 해당 사용자가 연결되는 파일과 프로세스가 많아지므로 해당 사용자의 권한이 높아집니다 .

3.3.1.2. 사용자 계정

사용자 계정은 실제 사람에게 할당되며 시스템에 로그인하고 사용하는 데 사용됩니다 . 시스템에 액세스하는 모든 사람은 고유한 사용자 계정을 가져야 합니다 . 이를 통해 관리자는 누가 어떤 작업을 하는지 파악할 수 있고 다른 사용자의 설정을 방해하는 것을 방지할 수 있습니다 .

각 사용자는 기본 셸 , 편집기 , 키 바인딩 및 언어 설정을 구성하여 시스템 사용에 맞게 자신만의 환경을 설정할 수 있습니다 .

FreeBSD 시스템의 모든 사용자 계정에는 특정 정보가 연관되어 있습니다 :

User name

사용자 이름은 login: 프롬프트에 입력합니다 . 각 사용자는 고유한 사용자 아이디를 가져야 합니다 . 유효한 사용자 아이디를 만들기 위한 여러 가지 규칙이 있으며 , 이 규칙은 passwd(5)에 문서화되어 있습니다 . 애플리케이션과의 이전 버전과의 호환성을 유지하려면 모두 소문자로 구성된 8자 이하의 사용자 아이디를 사용하는 것이 좋습니다 .

Password

각 계정에는 연결된 비밀번호가 있습니다 .

User ID(UID)

사용자 ID(UID)는 FreeBSD 시스템에서 사용자를 고유하게 식별하는 데 사용되는 번호입니다 . 사용자 이름을 지정할 수 있는 명령은 먼저 사용자 이름을 UID로 변환합니다 . 값이 크면 일부 소프트웨어와 호환성 문제가 발생할 수 있으므로 65535 미만의 UID를 사용하는 것이 좋습니다 .

Group ID(GID)

그룹 ID(GID)는 사용자가 속한 기본 그룹을 고유하게 식별하는 데 사용되는 번호입니다 . 그룹은 사용자의 UID가 아닌 GID를 기반으로 리소스에 대한 액세스를 제어하는 메커니즘입니다 . 이를 통해 일부 구성 파일의 크기를 크게 줄일 수 있으며 사용자가 둘 이상의 그룹의 구성원이 될 수 있습니다 . GID 가 높으면 일부 소프트웨어가 손상될 수 있으므로 65535 이하의 GID를 사용하는 것이 좋습니다 .

Login class

로그인 클래스는 그룹 메커니즘의 확장으로 , 시스템을 다양한 사용자에 맞게 조정할 때 추가적인 유연성을 제공합니다 . 로그인 클래스에 대한 자세한 내용은 로그인 클래스 구성하기에서 자세히 설명합니다 .

Password change time

기본적으로 비밀번호는 만료되지 않습니다 . 그러나 사용자별로 비밀번호 만료를 사용 설정하여 일정 시간이 경과한 후 일부 또는 모든 사용자가 비밀번호를 변경하도록 할 수 있습니다 .

Account expiration time

기본적으로 FreeBSD는 계정을 만료하지 않습니다 . 학교의 학생 계정과 같이 수명을 제한해야 하는 계정을 만들 때는 pw(8)을 사용하여 계정 만료 날짜를 지정하세요 . 만료 시간이 지나면 계정의 디렉터리와 파일은 남아 있지만 시스템에 로그인하는 데 해당 계정을 사용할 수 없습니다 .

User's full name

사용자 이름은 FreeBSD에 계정을 고유하게 식별하지만 반드시 사용자의 실명을 반영하지는 않습니다 . 이 정보는 댓글과 마찬가지로 공백 , 대문자를 포함할 수 있으며 8자 이상이어야 합니다 .

Home directory

홈 디렉터리는 시스템에서 디렉터리의 전체 경로입니다 . 사용자가 로그인할 때 사용자의 시작 디렉터리입니다 . 일반적인 규칙은 모든 사용자 홈 디렉터리를 `/home/username` 또는 `/usr/home/username` 아래에 배치하는 것입니다 . 각 사용자는 자신의 홈 디렉터리에 개인 파일과 하위 디렉터리를 저장합니다 .

User shell

셸은 시스템과 상호 작용하기 위한 사용자의 기본 환경을 제공합니다 . 셸에는 여러 가지 종류가 있으며 숙련된 사용자는 계정 설정에 반영할 수 있는 자신만의 기본 설정이 있을 것입니다 .

3.3.1.3. 수퍼유저 계정

root .

su(1) . wheel root .

make install exit .

Example 1.

```
% configure
% make
% su -
Password:
# make install
# exit
%
```

su(1) .

security/sudo .

3.3.2.

FreeBSD .

Table 1.

Command	Summary

adduser(8)	새 사용자를 추가하는 데 권장되는 명령줄 애플리케이션입니다 .
rmuser(8)	사용자 제거에 권장되는 명령줄 애플리케이션입니다 .
chpass(1)	사용자 데이터베이스 정보를 변경할 수 있는 유연한 도구입니다 .
passwd(1)	사용자 비밀번호를 변경하는 명령줄 도구입니다 .
pw(8)	사용자 계정의 모든 측면을 수정할 수 있는 강력하고 유연한 도구입니다 .
bsdconfig(8)	계정 관리가 지원되는 시스템 구성 유틸리티입니다 .

3.3.2.1. 사용자 추가하기

새 사용자를 추가하는 데 권장되는 프로그램은 adduser(8)입니다 . 이 프로그램은 새 사용자가 추가되면 자동으로 /etc/passwd 및 /etc/group을 업데이트합니다 . 또한 새 사용자를 위한 홈 디렉터리를 만들고 /usr/share/skel의 기본 구성 파일을 복사하며 선택적으로 새 사용자에게 환영 메시지를 메일로 보낼 수 있습니다 . 이 유틸리티는 수퍼유저로 실행해야 합니다 .

adduser(8) 유틸리티는 대화형이며 새 사용자 계정을 만드는 단계를 안내합니다 . FreeBSD에서 사용자 추가하기에서 볼 수 있듯이 필요한 정보를 입력하거나 Return 키를 눌러 대괄호 안에 표시된 기본값을 수락합니다 . 이 예에서는 사용자가 휠 그룹에 초대되어 su(1)을 가진 수퍼유저가 되었습니다 . 완료되면 유틸리티는 다른 사용자를 만들거나 종료하라는 메시지를 표시합니다 .

Exaple 2. FreeBSD에서 사용자 추가하기

```
# adduser
```

출력은 다음과 비슷하게 표시됩니다 :

```
Username: jru
Full name: J. Random User
Uid (Leave empty for default):
Login group [jru]:
Login group is jru. Invite jru into other groups? []: wheel
Login class [default]:
Shell (sh csh tcsh zsh nologin) [sh]: zsh
Home directory [/home/jru]:
Home directory permissions (Leave empty for default):
Use password-based authentication? [yes]:
Use an empty password? (yes/no) [no]:
Use a random password? (yes/no) [no]:
Enter password:
```

```
Enter password again:
Lock out the account after creation? [no]:
Username   : jru
Password   : ****
Full Name  : J. Random User
Uid        : 1001
Class      :
Groups     : jru wheel
Home       : /home/jru
Shell      : /usr/local/bin/zsh
Locked     : no
OK? (yes/no): yes
adduser: INFO: Successfully added (jru) to the user database.
Add another user? (yes/no): no
Goodbye!
```

Note

비밀번호를 입력할 때 에코가 울리지 않으므로 사용자 계정을 만들 때 비밀번호를 잘못 입력하지 않도록 주의하세요 .

3.3.2.2. 사용자 제거하기

시스템에서 사용자를 완전히 제거하려면 수퍼유저로 rmuser(8)를 실행합니다 . 이 명령은 다음 단계를 수행합니다 :

1. 사용자의 crontab(1) 항목이 있는 경우 해당 항목을 제거합니다 .
2. 사용자에게 속한 모든 at(1) 작업을 제거합니다 .
3. 사용자가 소유한 모든 프로세스에 SIGKILL 신호를 보냅니다 .
4. 시스템의 로컬 비밀번호 파일에서 사용자를 제거합니다 .
5. 실제 홈 디렉터리 경로의 심볼릭 링크 처리를 포함하여 사용자의 홈 디렉터리 (사용자가 소유한 경우) 를 제거합니다 .
6. /var/mail에서 사용자에게 속한 수신 메일 파일을 제거합니다 .
7. /tmp, /var/tmp 및 /var/tmp/vi.recover에서 사용자가 소유한 모든 파일을 제거합니다 .
8. /etc/group에서 해당 사용자 이름이 속한 모든 그룹에서 사용자 이름을 제거합니다 . (그룹이 비어 있고 그룹 이름이 사용자 이름과 같은 경우 그룹이 제거되며 , 이는 adduser(8)의 사용자별 고유

).

9. , .

rmuser(8) .

.

Example 3. rmuser

```
# rmuser jru
```

:

```
Matching password entry:
jru:*:1001:1001::0:0:J. Random User:/home/jru:/usr/local/bin/zsh
Is this the entry you wish to remove? y
Remove user's home directory (/home/jru)? y
Removing user (jru): mailspool home passwd.
```

3.3.2.3.

chpass(1) . .

chpass(1) .

Note
.

chpass jru . . jru . 6 . . chpass .

Example 4. chpass

```
# chpass
```

:

```
# Changing user database information for jru.
Login: jru
Password: *
Uid [#]: 1001
Gid [# or name]: 1001
Change [month day year]:
Expire [month day year]:
Class:
Home directory: /home/jru
Shell: /usr/local/bin/zsh
Full Name: J. Random User
Office Location:
Office Phone:
Home Phone:
Other information:
```

Example 5. 일반 사용자로 **chpass** 사용

```
#Changing user database information for jru.
Shell: /usr/local/bin/zsh
Full Name: J. Random User
Office Location:
Office Phone:
Home Phone:
Other information:
```

Note

명령 chfn(1) 및 chsh(1)는 chpass(1)에 대한 링크이며 , ypchpass(1), ypchfn(1) 및 ypchsh(1)도 마찬가지입니다 . NIS 지원은 자동으로 이루어지므로 명령 앞에 yp를 지정할 필요가 없습니다 . NIS를 구성하는 방법은 네트워크 서버에서 다루고 있습니다 .

3.3.2.4. 사용자 비밀번호 변경

모든 사용자는 passwd(1)를 사용하여 비밀번호를 쉽게 변경할 수 있습니다 . 실수나 무단 변경을 방지하기 위해 이 명령은 새 비밀번호를 설정하기 전에 사용자의 원래 비밀번호를 묻는 메시지를 표시합니다 :

선택적 사용자 이름 외에 다른 옵션을 전달하지 않으면 chpass(1)은 사용자 정보가 포함된 편집기를 표시합니다 . 사용자가 편집기를 종료하면 사용자 데이터베이스가 새 정보로 업데이트됩니다 .

Note

이 유틸리티는 수퍼유저로 실행하지 않는 한 편집기를 종료할 때 사용자의 비밀번호를 묻는 메시지를 표시합니다 .

수퍼유저에서 chpass jru를 입력했으며 이제 이 사용자에 대해 변경할 수 있는 필드를 보고 있습니다 . jru가 대신 이 명령을 실행하면 마지막 6개 필드만 표시되고 편집할 수 있습니다 . 이것은 다음에 표시됩니다 .

□□ II. □□□□ □□

□□ III. □□□ □□

□□ IV. □□□ □□

Chapter 34. 고급 네트워킹

34.1. 요약

이 장에서는 여러 가지 고급 네트워킹 주제를 다룹니다 .

이 장을 읽고 나면 다음 사항을 알게 될 것입니다 :

- 게이트웨이와 라우팅의 기본 사항 .
- USB 테더링을 설정하는 방법 .
- IEEE® 802.11 및 Bluetooth® 장치를 설정하는 방법 .
- FreeBSD가 브리지 역할을 하도록 하는 방법 .
- 네트워크 PXE 부팅을 설정하는 방법 .
- FreeBSD에서 공통 주소 중복 프로토콜 (CARP)의 기능을 활성화하고 활용하는 방법 .
- FreeBSD에서 여러 VLAN을 구성하는 방법 .
- 블루투스 헤드셋 구성하기 .

이 장을 읽기 전에 먼저 다음 사항을 이해해야 합니다 :

- **/etc/rc** 스크립트의 기본 사항을 이해합니다 .
- 기본적인 네트워크 용어를 숙지합니다 .
- FreeBSD의 기본 네트워크 구성을 이해합니다 (FreeBSD 네트워크).
- 새로운 FreeBSD 커널을 구성하고 설치하는 방법을 알 수 있습니다 (FreeBSD 커널 구성하기).
- 추가 서드파티 소프트웨어를 설치하는 방법을 알 수 있습니다 (애플리케이션 설치하기 : 패키지 및 포트).

34.2. 게이트웨이와 라우팅

라우팅은 한 시스템이 다른 시스템으로 가는 네트워크 경로를 찾을 수 있도록 하는 메커니즘입니다 . 라우팅은 ' 목적지 '와 '게이트웨이 '를 나타내는 정의된 주소 쌍입니다 . 경로는 지정된 목적지로 가려고 할 때 지정된 게이트웨이를 통해 패킷을 보내도록 지정합니다 . 목적지에는 개별 호스트 , 서브넷 , "default"의 세 가지 유형이 있습니다 . "default"는 다른 경로가 적용되지 않는 경우 사용됩니다 . 또한 게이트웨이에는 개별 호스트 , 인터페이스라고도 하는 링크 , 이더넷 하드웨어 (MAC) 주소의 세 가지 유형이 있습니다 . 알려진 경로는 라우팅 테이블에 저장됩니다 .

이 섹션에서는 라우팅 기본 사항에 대한 개요를 제공합니다 . 그런 다음 FreeBSD 시스템을 라우터로 구성하는 방법을 시연하고 몇 가지 문제 해결 팁을 제공합니다 .

34.2.1. 라우팅 기초

FreeBSD 시스템의 라우팅 테이블을 보려면 netstat(1)을 사용하세요 :

```
% netstat -r
Routing tables

Internet:
Destination      Gateway           Flags     Refs     Use     Netif Expire
default          outside-gw        UGS         37     418      em0
localhost        localhost         UH           0     181      lo0
test0            0:e0:b5:36:cf:4f  UHLW         5   63288      re0     77
10.20.30.255     link#1            UHLW         1    2421
example.com      link#1            UC           0       0
host1            0:e0:a8:37:8:1e   UHLW         3    4601      lo0
host2            0:e0:a8:37:8:1e   UHLW         0       5      lo0 =>
host2.example.com link#1           UC           0       0
224              link#1            UC           0       0
```

이 예제의 항목은 다음과 같습니다 :

default

이 표의 첫 번째 경로는 default 경로를 지정합니다 . 로컬 시스템이 원격 호스트에 연결해야 할 때 라우팅 테이블을 확인하여 알려진 경로가 존재하는지 확인합니다 . 원격 호스트가 테이블의 항목과 일치하면 시스템은 해당 항목에 지정된 인터페이스를 사용하여 연결할 수 있는지 확인합니다 .

목적지가 항목과 일치하지 않거나 알려진 경로가 모두 실패하면 시스템은 해당 항목을 기본 경로로 사용합니다 . 로컬 영역 네트워크에 있는 호스트의 경우 기본 경로의 Gateway 필드는 인터넷에 직접 연결되어 있는 시스템으로 설정됩니다 . 이 항목을 읽을 때 Flags 열에 게이트웨이가 사용 가능 (UG)임을 나타내는지 확인합니다 .

자체적으로 외부 세계의 게이트웨이로 작동하는 시스템의 기본 경로는 인터넷 서비스 제공업체 (ISP)의 게이트웨이 시스템이 됩니다 .

localhost

두 번째 경로는 localhost 경로입니다 . localhost의 Netif 열에 지정된 인터페이스는 루프백 장치라고도 하는 **lo0**입니다 . 이는 이 대상에 대한 모든 트래픽이 네트워크를 통해 전송되지 않고 내부에 있어야 함을 나타냅니다 .

MAC 주소

0:e0: 로 시작하는 주소는 MAC 주소입니다 . FreeBSD는 로컬 이더넷에서 호스트 (예제에서는 test0)를 자동으로 식별하고 이더넷 인터페이스를 통해 해당 호스트에 대한 경로 (**re0**)를 추가합니다 . 이 유형의 경로에는 Expire 열에서 볼 수 있는 시간 제한이 있으며 , 호스트가 특정 시간 동안 응답하지 않을 때 사용됩니다 . 이 경우 이 호스트에 대한 경로는 자동으로 삭제됩니다 . 이러한 호스트는 최단 경로 결정을 기반으로 로컬 호스트에 대한 경로를 계산하는 RIP(라우팅 정보 프로토콜)를 사용하여 식별됩니다 .

subnet

FreeBSD는 로컬 서브넷에 대한 서브넷 경로를 자동으로 추가합니다 . 이 예에서 10.20.30.255는 서브넷 10.20.30의 브로드캐스트 주소이고 example.com은 해당 서브넷과 연결된 도메인 이름입니다 . 지정 link#1은 머신의 첫 번째 이더넷 카드를 나타냅니다 .

로컬 네트워크 호스트와 로컬 서브넷은 routed(8)이라는 데몬에 의해 경로가 자동으로 구성됩니다 . 이 데몬이 실행되지 않으면 관리자가 정적으로 정의한 경로만 존재합니다 .

host

host1 줄은 이더넷 주소로 호스트를 나타냅니다 . 이 호스트는 보내는 호스트이므로 FreeBSD는 이더넷 인터페이스가 아닌 루프백 인터페이스 (**lo0**)를 사용한다는 것을 알고 있습니다 .

두 개의 host2 줄은 ifconfig(8)을 사용하여 생성된 별칭을 나타냅니다 . **lo0** 인터페이스 뒤의 ⇒ 기호는 루프백 주소 외에 별칭이 설정되었음을 나타냅니다 . 이러한 경로는 해당 별칭을 지원하는 호스트에만 표시되며 로컬 네트워크의 다른 모든 호스트에는 해당 경로에 대한 link#1 줄이 있습니다 .

224

마지막 줄 (대상 서브넷 224)은 멀티캐스팅을 처리합니다 .

각 경로의 다양한 속성은 Flags 열에서 볼 수 있습니다 . 흔히 볼 수 있는 라우팅 테이블 플래그에는 이러한 플래그 중 일부와 그 의미가 요약되어 있습니다 :

표 *1.* 흔히 볼 수 있는 라우팅 테이블 플래그

플래그	목적
U	경로가 활성 (up)입니다 .
H	경로 대상은 단일 호스트입니다 .
G	이 대상에 대한 모든 것을 이 게이트웨이로 보내면 이 게이트웨이에서 어디로 보낼지 알아냅니다 .

S	이 경로는 정적으로 구성되었습니다 .
C	이 경로를 기반으로 새 경로를 복제하여 머신이 연결할 수 있도록 합니다 . 이 유형의 경로는 일반적으로 로컬 네트워크에 사용됩니다 .
W	경로는 로컬 영역 네트워크 (복제) 경로를 기반으로 자동 구성되었습니다 .
L	경로에는 이더넷 (링크) 하드웨어에 대한 참조가 포함됩니다 .

FreeBSD 시스템에서는 기본 게이트웨이의 IP 주소를 지정하여 /etc/rc.conf에서 기본 경로를 정의할 수 있습니다 :

```
defaultrouter="10.20.30.1"
```

route를 사용하여 수동으로 경로를 추가할 수도 있습니다 :

```
$ route add default 10.20.30.1
```

수동으로 추가한 경로는 재부팅 후에도 유지되지 않습니다 . 네트워크 라우팅 테이블의 수동 조작에 대한 자세한 내용은 route(8)을 참조하세요 .

34.2.2. 정적 경로로 라우터 구성하기

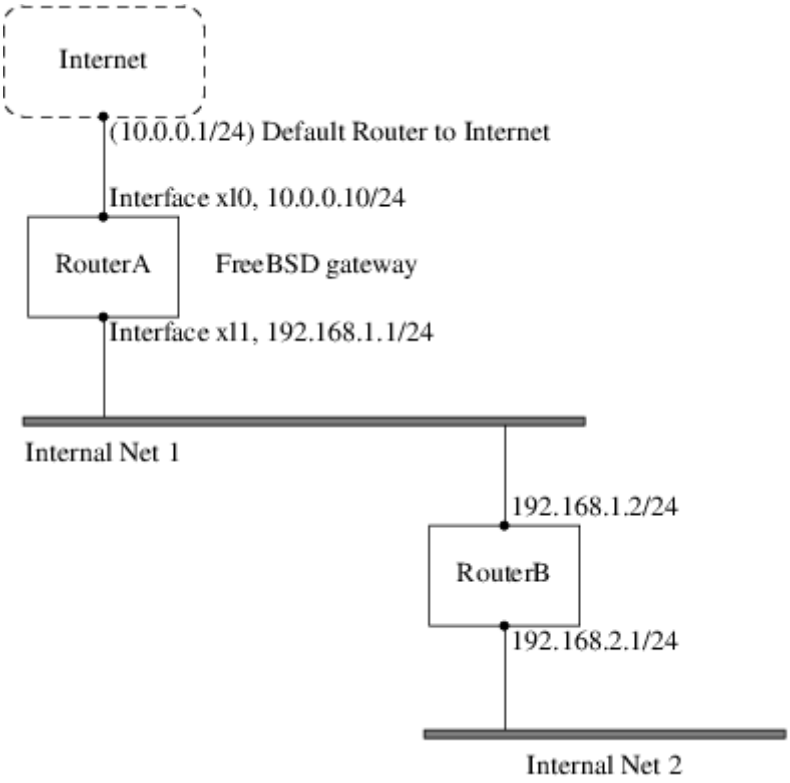
듀얼 홈 시스템인 경우 FreeBSD 시스템을 네트워크의 기본 게이트웨이 또는 라우터로 구성할 수 있습니다 . 듀얼 홈 시스템은 적어도 두 개의 서로 다른 네트워크에 상주하는 호스트입니다 . 일반적으로 각 네트워크는 별도의 네트워크 인터페이스에 연결되지만 , IP 앨리어싱을 사용하여 각각 다른 서브넷에 있는 여러 주소를 하나의 물리적 인터페이스에 바인딩할 수 있습니다 .

시스템이 인터페이스 간에 패킷을 전달하려면 FreeBSD를 라우터로 구성해야 합니다 . 인터넷 표준과 좋은 엔지니어링 관행으로 인해 FreeBSD 프로젝트는 이 기능을 기본적으로 활성화하지 않지만 **/etc/rc.conf**에 이 줄을 추가하여 부팅 시 시작되도록 구성할 수 있습니다 :

```
gateway_enable="YES"          # Set to YES if this host will be a gateway
```

지금 라우팅을 사용하려면 sysctl(8) 변수 net.inet.ip.forwarding을 1로 설정하세요 . 라우팅을 중지하려면 이 변수를 0으로 재설정하세요 .

라우터의 라우팅 테이블에는 다른 네트워크에 도달하는 방법을 알 수 있도록 추가 경로가 필요합니다 . 경로는 정적 경로를 사용하여 수동으로 추가하거나 라우팅 프로토콜을 사용하여 경로를 자동으로 학습할 수 있습니다 . 정적 경로는 소규모 네트워크에 적합하며 이 섹션에서는 소규모 네트워크에 정적 라우팅 항목을 추가하는



방법을 설명합니다 .

> **Note**
>
> 대규모 네트워크의 경우 , 정적 경로는 빠르게 확장할 수 없게 됩니다 . FreeBSD는 라우팅 프로토콜인 RIP, 버전 1과 2, IRDP를 제공하는 표준 BSD 라우팅 데몬 routed(8)와 함께 제공됩니다 . BGP 및 OSPF 라우팅 프로토콜에 대한 지원은 net/quagga 패키지 또는 포트를 사용하여 설치할 수 있습니다 .

다음 네트워크를 생각해 보세요 :

이 시나리오에서 라우터 A는 나머지 인터넷에 대한 라우터 역할을 하는 FreeBSD 머신입니다 . 기본 경로가 10.0.0.1로 설정되어 외부 세계와 연결할 수 있습니다 . 라우터 B는 이미 192.168.1.1을 기본 게이트웨이로 사용하도록 구성되어 있습니다 .

정적 경로를 추가하기 전 라우터 A의 라우팅 테이블은 다음과 같습니다 :

```
$ netstat -nr
Routing tables

Internet:
Destination       Gateway           Flags    Refs      Use  Netif  Expire
default           10.0.0.1          UGS         0    49378    xl0
127.0.0.1         127.0.0.1         UH          0        6    lo0
```

```
10.0.0.0/24        link#1            UC         0        0    xl0
192.168.1.0/24     link#2            UC         0        0    xl1
```

현재 라우팅 테이블을 사용하면 라우터 A에 192.168.2.0/24 네트워크에 대한 경로가 없습니다 . 다음 명령은 192.168.1.2를 다음 홉으로 사용하여 라우터 A의 라우팅 테이블에 Internal Net 2 네트워크를 추가합니다 :

```
$ route add -net 192.168.2.0/24 192.168.1.2
```

이제 라우터 A는 192.168.2.0/24 네트워크의 모든 호스트에 연결할 수 있습니다 . 그러나 FreeBSD 시스템이 재부팅되면 라우팅 정보가 유지되지 않습니다 . 정적 경로를 영구적으로 유지해야 하는 경우 **/etc/rc.conf**에 추가하세요 :

```
# Add Internal Net 2 as a persistent static route
static_routes="internalnet2"
route_internalnet2="-net 192.168.2.0/24 192.168.1.2"
```

static_routes 구성 변수는 공백으로 구분된 문자열 목록으로 , 각 문자열은 경로 이름을 참조합니다 . route_internalnet2 변수에는 해당 경로 이름에 대한 정적 경로가 포함됩니다 .

static_routes에 둘 이상의 문자열을 사용하면 여러 개의 정적 경로가 생성됩니다 . 다음은 192.168.0.0/24 및 192.168.1.0/24 네트워크에 대한 정적 경로를 추가하는 예제입니다 :

```
static_routes="net1 net2"
route_net1="-net 192.168.0.0/24 192.168.0.1"
route_net2="-net 192.168.1.0/24 192.168.1.1"
```

34.2.3. 문제 해결

주소 공간이 네트워크에 할당되면 서비스 제공업체는 네트워크의 모든 트래픽이 해당 사이트의 링크로 전송되도록 라우팅 테이블을 구성합니다 . 그렇다면 외부 사이트는 네트워크의 ISP에 패킷을 보낼지 어떻게 알 수 있을까요 ?

할당된 모든 주소 공간을 추적하고 인터넷 백본 또는 전국 및 전 세계 인터넷 트래픽을 전송하는 주요 간선 회선에 대한 연결 지점을 정의하는 시스템이 있습니다 . 각 백본 머신에는 특정 네트워크의 트래픽을 특정 백본 캐리어로 , 그리고 거기에서 특정 네트워크에 도달할 때까지 서비스 제공업체 체인을 따라 내려가는 마스터 테이블 세트의 사본이 있습니다 .

서비스 제공업체는 백본 사이트에 자신이 사이트의 연결 지점 , 즉 내부 경로임을 알리는 것이 서비스 제공업체의 임무입니다 . 이를 경로 전파라고 합니다 .

간혹 경로 전파에 문제가 발생하여 일부 사이트에 연결할 수 없는 경우가 있습니다 . 라우팅이 고장난 위치를 파악하는 데 가장 유용한 명령은 traceroute 입니다 . ping이 실패할 때 유용합니다 .

traceroute를 사용할 때는 연결할 원격 호스트의 주소를 포함하세요 . 출력에는 시도 경로를 따라 게이트웨이 호스트가 표시되며 , 결국 대상 호스트에 도달하거나 연결 부족으로 인해 종료됩니다 . 자세한 내용은 traceroute(8)을 참조하세요 .

34.2.4. 멀티캐스트 고려 사항

FreeBSD는 기본적으로 멀티캐스트 애플리케이션과 멀티캐스트 라우팅을 모두 지원합니다 . 멀티캐스트 애플리케이션은 FreeBSD에서 실행하기 위해 특별한 구성이 필요하지 않습니다 . 멀티캐스트 라우팅을 지원하려면 다음 옵션을 커스텀 커널에 컴파일해야 합니다 :

```
options MROUTING
```

멀티캐스트 라우팅 데몬인 mrouted는 net/mrouted 패키지 또는 포트를 사용하여 설치할 수 있습니다 . 이 데몬은 DVMRP 멀티캐스트 라우팅 프로토콜을 구현하며 터널과 DVMRP를 설정하기 위해 **/usr/local/etc/mrouted.conf**를 편집하여 구성합니다 . mrouted를 설치하면 map-mbone 및 mrinfo와 관련 매뉴얼 페이지도 설치됩니다 . 구성 예제는 이를 참조하세요 .

> **Note**
> 많은 멀티캐스트 설치에서 DVMRP는 대부분 PIM 프로토콜로 대체되었습니다 . 자세한 내용은 pim(4)를 참조하세요 .

34.3. 가상 호스트

하나의 서버가 네트워크에 여러 서버로 표시되는 가상 사이트 호스팅은 FreeBSD의 일반적인 용도입니다 . 이는 단일 인터페이스에 여러 네트워크 주소를 할당하여 이루어집니다 .

주어진 네트워크 인터페이스에는 하나의 "실제 " 주소가 있으며 , "별칭 " 주소는 얼마든지 가질 수 있습니다 . 이러한 별칭은 일반적으로 이 예에서 보는 것처럼 **/etc/rc.conf**에 별칭 항목을 배치하여 추가합니다 :

```
$ sysrc ifconfig_fxp0_alias0="inet xxx.xxx.xxx.xxx netmask xxx.xxx.xxx.xxx"
```

alias 항목은 alias0, alias1 등과 같은 일련번호를 사용하여 alias0으로 시작해야 합니다 . 구성 프로세스는 첫 번째 누락된 번호에서 중지됩니다 .

별칭 넷마스크 계산은 중요합니다 . 주어진 인터페이스에 대해 네트워크의 넷마스크를 올바르게 나타내는 주소가 하나 있어야 합니다 . 이 네트워크에 속하는 다른 모든 주소는 255.255.255.255 또는 0xffffffff로 표현되는 모두 1로 구성된 넷마스크를 가져야 합니다 .

예를 들어 , fxp0 인터페이스가 두 개의 네트워크에 연결되어 있는 경우를 생각해 보겠습니다 : 넷마스크가 255.255.255.0인 10.1.1.0과 넷마스크가 255.255.255.240인 202.0.75.16입니다 . 시스템은 10.1.1.1 ~ 10.1.1.5 및 202.0.75.17 ~ 202.0.75.20 범위에 나타나도록 구성해야 합니다 . 주어진 네트워크 범위의 첫 번째 주소에만 실제 넷마스크가 있어야 합니다 . 나머지 모든 주소 (10.1.1.2 ~ 10.1.1.5 및 202.0.75.18 ~ 202.0.75.20)는 255.255.255.255의 넷마스크로 구성해야 합니다 .

다음 **/etc/rc.conf** 항목은 이 시나리오에 맞게 어댑터를 올바르게 구성합니다 :

```
$ sysrc ifconfig_fxp0="inet 10.1.1.1 netmask 255.255.255.0"
$ sysrc ifconfig_fxp0_alias0="inet 10.1.1.2 netmask 255.255.255.255"
$ sysrc ifconfig_fxp0_alias1="inet 10.1.1.3 netmask 255.255.255.255"
$ sysrc ifconfig_fxp0_alias2="inet 10.1.1.4 netmask 255.255.255.255"
$ sysrc ifconfig_fxp0_alias3="inet 10.1.1.5 netmask 255.255.255.255"
$ sysrc ifconfig_fxp0_alias4="inet 202.0.75.17 netmask 255.255.255.240"
$ sysrc ifconfig_fxp0_alias5="inet 202.0.75.18 netmask 255.255.255.255"
$ sysrc ifconfig_fxp0_alias6="inet 202.0.75.19 netmask 255.255.255.255"
$ sysrc ifconfig_fxp0_alias7="inet 202.0.75.20 netmask 255.255.255.255"
```

이를 더 간단하게 표현하는 방법은 공백으로 구분된 IP 주소 범위 목록을 사용하는 것입니다 . 첫 번째 주소에는 지정된 서브넷 마스크가 지정되고 추가 주소에는 255.255.255.255의 서브넷 마스크가 지정됩니다 .

```
$ sysrc ifconfig_fxp0_aliases="inet 10.1.1.1-5/24 inet 202.0.75.17-20/28"
```

34.4. 무선 고급 인증

FreeBSD는 무선 네트워크에 연결하는 다양한 방법을 지원합니다 . 이 섹션에서는 무선 네트워크에 대한 고급 인증을 수행하는 방법을 설명합니다 .

무선 네트워크에 연결하고 기본 인증을 하려면 네트워크 장의 무선 네트워크에 연결 및 인증 섹션에서 방법을 설명합니다 .

34.4.1. EAP-TLS를 사용한 WPA

두 번째 WPA 사용 방법은 802.1X 백엔드 인증 서버를 사용하는 것입니다 . 이 경우 보안 수준이 낮은 WPA 개인과 구분하기 위해 WPA를 WPA 엔터프라이즈라고 합니다 . WPA Enterprise의 인증은 EAP(확장 인증 프로토콜)를 기반으로 합니다 .

EAP에는 암호화 방법이 제공되지 않습니다 . 대신 EAP는 암호화된 터널 안에 내장되어 있습니다 . EAP 인증 방법은 여러 가지가 있지만 EAP-TLS, EAP-TTLS 및 EAP-PEAP가 가장 일반적입니다 .

EAP-TLS(전송 계층 보안을 사용한 EAP)는 Wi-Fi Alliance에서 인증한 최초의 EAP 방법이기 때문에 잘 지원되는 무선 인증 프로토콜입니다 . EAP-TLS를 실행하려면 모든 시스템에 설치된 CA(인증 기관)의 인증서 , 인증 서버용 서버 인증서 , 각 무선 클라이언트에 대한 클라이언트 인증서 등 세 개의 인증서가 필요합니다 . 이 EAP 방식에서는 인증 서버와 무선 클라이언트가 각각의 인증서를 제시하여 서로를 인증한 다음 , 이 인증서가 조직의 CA에 의해 서명되었는지 확인합니다 .

이전과 마찬가지로 구성은 **/etc/wpa_supplicant.conf**를 통해 이루어집니다 :

```
network={
  ssid="freebsdap"                      - 1
  proto=RSN                             - 2
  key_mgmt=WPA-EAP                      - 3
  eap=TLS                               - 4
  identity="loader"                     - 5
  ca_cert="/etc/certs/cacert.pem"       - 6
  client_cert="/etc/certs/clientcert.pem"  - 7
  private_key="/etc/certs/clientkey.pem"   - 8
  private_key_passwd="freebsdmallclient"   - 9
}
```

1. 이 필드는 네트워크 이름 (SSID)을 나타냅니다 .
2. 이 예에서는 WPA2라고도 하는 RSN IEEE® 802.11i 프로토콜을 사용합니다 .
3. key_mgmt 줄은 사용할 키 관리 프로토콜을 나타냅니다 . 이 예에서는 EAP 인증을 사용하는 WPA 입니다 .
4. 이 필드는 연결에 대한 EAP 방법을 나타냅니다 .
5. identity 필드에는 EAP의 ID 문자열이 포함됩니다 .
6. ca_cert 필드는 CA 인증서 파일의 경로명을 나타냅니다 . 이 파일은 서버 인증서를 확인하는 데 필요합니다 .
7. client_cert 줄은 클라이언트 인증서 파일의 경로명을 제공합니다 . 이 인증서는 네트워크의 각 무선 클라이언트마다 고유합니다 .

8. private_key 필드는 클라이언트 인증서 개인 키 파일의 경로 이름입니다 .

9. private_key_passwd 필드에는 개인 키의 암호 구문이 포함됩니다 .

그런 다음 **/etc/rc.conf**에 다음 줄을 추가합니다 :

```
wlans_ath0="wlan0"
ifconfig_wlan0="WPA DHCP"
```

다음 단계는 인터페이스를 불러오는 것입니다 :

```
$ service netif start
Starting wpa_supplicant.
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 7
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 15
DHCPACK from 192.168.0.20
bound to 192.168.0.254 -- renewal in 300 seconds.
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    ether 00:11:95:d5:43:62
    inet 192.168.0.254 netmask 0xffffff00 broadcast 192.168.0.255
    media: IEEE 802.11 Wireless Ethernet DS/11Mbps mode 11g
    status: associated
    ssid freebsdap channel 1 (2412 Mhz 11g) bssid 00:11:95:c3:0d:ac
    country US ecm authmode WPA2/802.11i privacy ON deftxkey UNDEF
    AES-CCM 3:128-bit txpower 21.5 bmiss 7 scanvalid 450 bgscan
    bgscanintvl 300 bgscanidle 250 roam:rssi 7 roam:rate 5 protmode CTS
    wme burst roaming MANUAL
```

wpa_supplicant(8) 및 ifconfig(8)를 사용하여 인터페이스를 수동으로 불러올 수도 있습니다 .

34.4.2. EAP-TTLS를 사용한 WPA

EAP-TLS를 사용하면 인증 서버와 클라이언트 모두 인증서가 필요합니다 . EAP-TTLS를 사용하면 클라이언트 인증서는 선택 사항입니다 . 이 방법은 방문자에게 클라이언트 측 인증서가 없는 경우에도 안전한 SSL 터널을 생성하는 웹 서버와 유사합니다 . EAP-TTLS는 인증 데이터의 안전한 전송을 위해 암호화된 TLS 터널을 사용합니다 .

필요한 구성은 **/etc/wpa_supplicant.conf**에 추가할 수 있습니다 :

```
network={
  ssid="freebsdap"
  proto=RSN
  key_mgmt=WPA-EAP
  eap=TTLS                    - 1
  identity="test"             - 2
  password="test"             - 3
  ca_cert="/etc/certs/cacert.pem"   - 4
  phase2="auth=MD5"           - 5
}
```

1. 이 필드는 연결에 대한 EAP 방법을 지정합니다 .
2. ID 필드에는 암호화된 TLS 터널 내부의 EAP 인증을 위한 ID 문자열이 포함됩니다 .
3. password 필드에는 EAP 인증을 위한 암호가 포함됩니다 .
4. ca_cert 필드는 CA 인증서 파일의 경로명을 나타냅니다 . 이 파일은 서버 인증서를 확인하는 데 필요합니다 .
5. 이 필드는 암호화된 TLS 터널에서 사용되는 인증 방법을 지정합니다 . 이 예에서는 MD5-Challenge 가 포함된 EAP가 사용됩니다 . "내부 인증 " 단계는 종종 "phase2"라고 합니다 .

다음으로 **/etc/rc.conf**에 다음 줄을 추가합니다 :

```
wlans_ath0="wlan0"
ifconfig_wlan0="WPA DHCP"
```

다음 단계는 인터페이스를 불러오는 것입니다 :

```
$ service netif start
Starting wpa_supplicant.
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 7
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 15
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 21
DHCPACK from 192.168.0.20
bound to 192.168.0.254 -- renewal in 300 seconds.
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
      ether 00:11:95:d5:43:62
      inet 192.168.0.254 netmask 0xffffff00 broadcast 192.168.0.255
      media: IEEE 802.11 Wireless Ethernet DS/11Mbps mode 11g
```

```
status: associated
ssid freebsdap channel 1 (2412 Mhz 11g) bssid 00:11:95:c3:0d:ac
country US ecm authmode WPA2/802.11i privacy ON deftxkey UNDEF
AES-CCM 3:128-bit txpower 21.5 bmiss 7 scanvalid 450 bgscan
bgscanintvl 300 bgscanidle 250 roam:rssi 7 roam:rate 5 protmode CTS
wme burst roaming MANUAL
```

34.4.3. EAP-PEAP를 사용한 WPA

<Note>
PEAPv0/EAP-MSCHAPv2는 가장 일반적인 PEAP 방식입니다. 이 장에서는 이 방법을 지칭할 때 PEAP라는 용어를 사용합니다.

Protected EAP(PEAP)는 EAP-TTLS의 대안으로 설계되었으며 EAP-TLS 다음으로 가장 많이 사용되는 EAP 표준입니다. 운영 체제가 혼합된 네트워크에서는 PEAP가 EAP-TLS 다음으로 가장 많이 지원되는 표준이어야 합니다.

PEAP는 서버 측 인증서를 사용하여 클라이언트와 인증 서버 사이에 암호화된 TLS 터널을 생성하여 클라이언트를 인증하고 이후 인증 정보 교환을 보호한다는 점에서 EAP-TTLS와 유사합니다. PEAP 인증은 사용자 이름을 투명하게 브로드캐스트하고 암호만 암호화된 TLS 터널로 전송한다는 점에서 EAP-TTLS와 다릅니다. EAP-TTLS는 사용자 이름과 비밀번호 모두에 TLS 터널을 사용합니다.

EAP-PEAP 관련 설정을 구성하려면 **/etc/wpa_supplicant.conf**에 다음 줄을 추가합니다:

```
network={
  ssid="freebsdap"
  proto=RSN
  key_mgmt=WPA-EAP
  eap=PEAP                    - 1
  identity="test"             - 2
  password="test"              - 3
  ca_cert="/etc/certs/cacert.pem"   - 4
  phase1="peaplabel=0"          - 5
  phase2="auth=MSCHAPV2"         - 6
}
```

1. 이 필드는 연결에 대한 EAP 방법을 지정합니다.
2. ID 필드에는 암호화된 TLS 터널 내부의 EAP 인증을 위한 ID 문자열이 포함됩니다.
3. password 필드에는 EAP 인증을 위한 암호가 포함됩니다.

4. ca_cert 필드는 CA 인증서 파일의 경로명을 나타냅니다 . 이 파일은 서버 인증서를 확인하는 데 필요합니다 .

5. 이 필드에는 인증의 첫 번째 단계인 TLS 터널에 대한 매개변수가 포함되어 있습니다 . 사용하는 인증 서버에 따라 인증에 사용할 특정 레이블을 지정합니다 . 대부분의 경우 , 라벨은 peaplabel=0을 사용하여 설정하는 "클라이언트 EAP 암호화 "가 됩니다 . 자세한 내용은 wpa_supplicant.conf(5) 에서 확인할 수 있습니다 .

6. 이 필드는 암호화된 TLS 터널에서 사용되는 인증 프로토콜을 지정합니다 . PEAP의 경우 auth=MSCHAPV2입니다 .

/etc/rc.conf에 다음을 추가합니다 :

```
wlans_ath0="wlan0"
ifconfig_wlan0="WPA DHCP"
```

그런 다음 인터페이스를 불러옵니다 :

```
$ service netif start
Starting wpa_supplicant.
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 7
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 15
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 21
DHCPACK from 192.168.0.20
bound to 192.168.0.254 -- renewal in 300 seconds.
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    ether 00:11:95:d5:43:62
    inet 192.168.0.254 netmask 0xffffff00 broadcast 192.168.0.255
    media: IEEE 802.11 Wireless Ethernet DS/11Mbps mode 11g
    status: associated
    ssid freebsdap channel 1 (2412 Mhz 11g) bssid 00:11:95:c3:0d:ac
    country US ecm authmode WPA2/802.11i privacy ON deftxkey UNDEF
    AES-CCM 3:128-bit txpower 21.5 bmiss 7 scanvalid 450 bgscan
    bgscanintvl 300 bgscanidle 250 roam:rssi 7 roam:rate 5 protmode CTS
    wme burst roaming MANUAL
```

34.5. 무선 Ad-hoc 모드

애드혹 모드라고도 하는 IBSS 모드는 지점 간 연결을 위해 설계되었습니다 . 예를 들어 , 머신 A와 B 사이에 애드혹 네트워크를 설정하려면 두 개의 IP 주소와 하나의 SSID를 선택합니다 .

A에서 :

```
$ ifconfig wlan0 create wlandev ath0 wlanmode adhoc
$ ifconfig wlan0 inet 192.168.0.1 netmask 255.255.255.0 ssid freebsdap
$ ifconfig wlan0
  wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        ether 00:11:95:c3:0d:ac
        inet 192.168.0.1 netmask 0xffffff00 broadcast 192.168.0.255
        media: IEEE 802.11 Wireless Ethernet autoselect mode 11g <adhoc>
        status: running
        ssid freebsdap channel 2 (2417 Mhz 11g) bssid 02:11:95:c3:0d:ac
        country US ecm authmode OPEN privacy OFF txpower 21.5 scanvalid 60
        protmode CTS wme burst
```

adhoc 매개변수는 인터페이스가 IBSS 모드에서 실행 중임을 나타냅니다 .

이제 B가 A를 감지할 수 있어야 합니다 :

```
$ ifconfig wlan0 create wlandev ath0 wlanmode adhoc
$ ifconfig wlan0 up scan
  SSID/MESH ID    BSSID              CHAN RATE   S:N     INT CAPS
  freebsdap       02:11:95:c3:0d:ac    2   54M -64:-96  100 IS   WME
```

출력의 I는 A가 임시 모드임을 확인합니다 . 이제 B를 다른 IP 주소로 구성합니다 :

```
$ ifconfig wlan0 inet 192.168.0.2 netmask 255.255.255.0 ssid freebsdap
$ ifconfig wlan0
  wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        ether 00:11:95:d5:43:62
        inet 192.168.0.2 netmask 0xffffff00 broadcast 192.168.0.255
        media: IEEE 802.11 Wireless Ethernet autoselect mode 11g <adhoc>
        status: running
        ssid freebsdap channel 2 (2417 Mhz 11g) bssid 02:11:95:c3:0d:ac
        country US ecm authmode OPEN privacy OFF txpower 21.5 scanvalid 60
        protmode CTS wme burst
```

이제 A와 B는 모두 정보를 교환할 준비가 되었습니다 .

34.5.1. FreeBSD 호스트 액세스 포인트

□□ V. □□

FreeBSD 과거 핸드북 번역판

최정호님께서 제공해 주신 것으로 알고 있습니다 .

과거에 번역했던 한글 번역판이라고 합니다 .

FreeBSD Korean.pdf