

Table of Contents

- 1. 전체 빌드 자동화 스크립트
- 2. 기능별 자동화 설명
 - 1. 환경 준비
 - 2. 베이스 ISO 다운로드
 - 3. ISO 파일 추출
 - 4. Chroot 환경 생성
 - 5. 패키지 설치 및 사용자 정의
 - 6. ISO 재포장
 - 7. 소스코드 공개
 - 8. 정리
- 3. 사용 방법
- 4. 주요 참고사항

"Lil Nut" 배포판 제작을 자동화하는 추가 스크립트를 작성해보겠습니다. 아래 스크립트는 배포판 생성 과정에서 필요한 다양한 작업을 자동으로 처리하도록 설계되었습니다.

1. 전체 빌드 자동화 스크립트

아래 스크립트는 ISO 빌드, 소스코드 관리, 라이선스 파일 생성, 그리고 사용자 정의 사항 반영을 모두 자동화합니다.

```
bash
```

```
#!/bin/bash
```

```
# Lil Nut Linux Distribution Build Script
```

```
# Author: Lil Nut Team
```

```
# Description: Automates the process of building the Lil Nut Linux distribution.
```

```
set -e
```

```
# === Variables ===
BUILD_DIR=~/.lilnut-build
CHROOT_DIR=$BUILD_DIR/chroot
ISO_DIR=$BUILD_DIR/iso
OUTPUT_DIR=$BUILD_DIR/output
SOURCE_REPO="https://github.com/<YourUsername>/lilnut-linux"
UBUNTU_BASE_ISO="ubuntu-18.04.6-desktop-amd64.iso"
ISO_NAME="LiINut.iso"
LICENSE_FILE="LICENSE"
SCRIPTS_DIR="$CHROOT_DIR/usr/local/bin"

# === Functions ===

# Step 1: Prepare build environment
prepare_environment() {
    echo "[INFO] Preparing build environment..."
    mkdir -p $BUILD_DIR $CHROOT_DIR $ISO_DIR $OUTPUT_DIR
}

# Step 2: Download base ISO
download_base_iso() {
    if [ ! -f $BUILD_DIR/$UBUNTU_BASE_ISO ]; then
        echo "[INFO] Downloading Ubuntu base ISO..."
        wget -O $BUILD_DIR/$UBUNTU_BASE_ISO
        "https://releases.ubuntu.com/18.04/$UBUNTU_BASE_ISO"
    else
        echo "[INFO] Base ISO already exists. Skipping download."
    fi
}

# Step 3: Extract ISO
extract_iso() {
    echo "[INFO] Extracting Ubuntu ISO..."
    sudo mount -o loop $BUILD_DIR/$UBUNTU_BASE_ISO $ISO_DIR/mount
    rsync -a $ISO_DIR/mount/ $ISO_DIR/
    sudo umount $ISO_DIR/mount
    rmdir $ISO_DIR/mount
}

# Step 4: Create chroot environment
setup_chroot() {
    echo "[INFO] Setting up chroot environment..."
    sudo debootstrap --arch=amd64 bionic $CHROOT_DIR
    http://archive.ubuntu.com/ubuntu/
}

# Step 5: Customize chroot (install packages)
customize_chroot() {
    echo "[INFO] Customizing chroot environment..."
    sudo chroot $CHROOT_DIR bash -c "
        apt update
        apt install -y lxqt awesome xorg busybox wine
        apt clean
    "
```

```
}
```

Step 6: Add custom scripts and files

```
add_custom_files() {  
    echo "[INFO] Adding custom files and scripts..."  
    mkdir -p $SCRIPTS_DIR  
    echo "echo 'Welcome to Lil Nut Linux!' > /etc/motd" > $SCRIPTS_DIR/setup.sh  
    chmod +x $SCRIPTS_DIR/setup.sh  
  
    echo "[INFO] Adding LICENSE file..."  
    cat > $CHROOT_DIR/usr/share/doc/$LICENSE_FILE << EOF  
Lil Nut Linux Distribution  
Based on Ubuntu 18.04, licensed under GNU GPL v2 or later.  
Custom modifications are under MIT License.  
EOF  
}
```

Step 7: Repack ISO

```
repack_iso() {  
    echo "[INFO] Repacking ISO..."  
    cd $ISO_DIR  
    mkisofs -o $OUTPUT_DIR/$ISO_NAME \  
        -b isolinux/isolinux.bin -c isolinux/boot.cat \  
        -no-emul-boot -boot-load-size 4 -boot-info-table \  
        -J -R -V "LilNut" \  
        $ISO_DIR  
    cd -  
}
```

Step 8: Publish source code

```
publish_source_code() {  
    echo "[INFO] Publishing source code to GitHub..."  
    git clone $SOURCE_REPO $BUILD_DIR/source  
    cp -r $CHROOT_DIR $BUILD_DIR/source/  
    cd $BUILD_DIR/source  
    git add .  
    git commit -m "Update source code"  
    git push origin main  
    cd -  
}
```

Step 9: Clean up

```
cleanup() {  
    echo "[INFO] Cleaning up..."  
    sudo umount $CHROOT_DIR/proc || true  
    sudo umount $CHROOT_DIR/sys || true  
    sudo umount $CHROOT_DIR/dev || true  
}
```

=== Main Script Execution ===

```
prepare_environment  
download_base_iso  
extract_iso  
setup_chroot  
customize_chroot
```

```
add_custom_files
repack_iso
publish_source_code
cleanup
```

```
echo "[INFO] Build process completed. ISO is located at $OUTPUT_DIR/$ISO_NAME"
```

2. 기능별 자동화 설명

1. 환경 준비

- 작업 디렉토리 생성.
- 필요한 디렉토리(`BUILD_DIR` , `CHROOT_DIR` , `ISO_DIR` , `OUTPUT_DIR`) 초기화.

2. 베이스 ISO 다운로드

- Ubuntu 18.04 ISO를 다운로드합니다.
- 기존 ISO가 있을 경우 건너뛴니다.

3. ISO 파일 추출

- 다운로드한 ISO를 추출하고 복사합니다.

4. Chroot 환경 생성

- `debootstrap` 을 사용해 최소한의 Ubuntu 환경을 설정합니다.

5. 패키지 설치 및 사용자 정의

- `lxqt` , `awesomewm` , `busybox` , `wine` 설치.
- `/etc/motd` 와 같은 사용자 정의 메시지 설정.

6. ISO 재포장

- `mkisofs` 를 사용해 커스터마이징된 ISO를 생성.

7. 소스코드 공개

- GitHub 저장소와 연동해 소스코드 업로드.

8. 정리

- ISO 제작 과정에서 사용된 임시 마운트 및 디렉토리 해제.

3. 사용 방법

- 스크립트 저장 위 스크립트를 `build-lilnut.sh` 로 저장합니다.
- 실행 권한 부여

```
bash
```

```
chmod +x build-lilnut.sh
```

3. 스크립트 실행

```
bash
```

```
sudo ./build-lilnut.sh
```

4. 주요 참고사항

- 의존성 확인:

debootstrap , mkisofs , git , wget 등 필요한 패키지가 설치되어 있어야 합니다.

```
bash
```

```
sudo apt install debootstrap mkisofs git wget
```

- GitHub 설정:

GitHub 저장소를 미리 생성하고, SOURCE_REPO 변수에 URL을 설정해야 합니다.

추가적으로 필요하거나 수정해야 할 부분이 있다면 알려주세요! 😊

```
#!/bin/bash
# Full Linux Distro Build Automation Script

set -e

# Variables
DISTRO_NAME="CustomLinux"
VERSION="1.0"
USERNAME="customuser"
PASSWORD="password123"
HOSTNAME="custom-linux"
ISO_OUTPUT="$DISTRO_NAME-$VERSION.iso"
WORKDIR=~/.custom-linux
CHROOTDIR="$WORKDIR/chroot"
ISODIR="$WORKDIR/iso"

# Ensure required tools are installed
echo "[INFO] Installing required tools..."
sudo apt update
sudo apt install -y debootstrap squashfs-tools xorriso grub-pc-bin grub-efi-amd64-bin mtools
genisoimage

# Step 1: Prepare directories
```

```
echo "[INFO] Preparing directories..."
mkdir -p $CHROOTDIR $ISODIR/boot/grub $ISODIR/casper $ISODIR/EFI $ISODIR/isolinux
```

Step 2: Create base system with debootstrap

```
echo "[INFO] Creating base system with debootstrap..."
sudo debootstrap --arch=amd64 focal $CHROOTDIR http://archive.ubuntu.com/ubuntu
```

Step 3: Configure chroot environment

```
echo "[INFO] Configuring chroot environment..."
sudo mount --bind /dev $CHROOTDIR/dev
sudo mount --bind /proc $CHROOTDIR/proc
sudo mount --bind /sys $CHROOTDIR/sys
```

```
sudo chroot $CHROOTDIR /bin/bash << EOF
set -e
echo "[INFO] Setting up the chroot environment..."
```

Basic settings

```
echo "$HOSTNAME" > /etc/hostname
echo "127.0.0.1 localhost" > /etc/hosts
echo "127.0.1.1 $HOSTNAME" >> /etc/hosts
```

Install essential packages

```
apt update
apt install -y linux-image-generic systemd-sysv busybox lxqt awesome wine sudo grub2
network-manager casper
```

Configure user account

```
useradd -m -s /bin/bash -G sudo,audio,video,plugdev $USERNAME
echo "$USERNAME:$PASSWORD" | chpasswd
echo "root:$PASSWORD" | chpasswd
```

Configure default user files

```
echo "export PS1='\u@\h:\w\$ '" > /home/$USERNAME/.bashrc
echo "Welcome to $DISTRO_NAME!" > /home/$USERNAME/welcome.txt
chown $USERNAME:$USERNAME /home/$USERNAME/.bashrc
/home/$USERNAME/welcome.txt
```

Clean up

```
apt clean
rm -rf /var/lib/apt/lists/*
EOF
```

```
sudo umount $CHROOTDIR/{dev,proc,sys}
```

Step 4: Create squashfs file

```
echo "[INFO] Creating squashfs file..."
sudo mksquashfs $CHROOTDIR $ISODIR/casper/filesystem.squashfs -comp xz -e boot
```

Step 5: Configure GRUB bootloader

```
echo "[INFO] Configuring GRUB bootloader..."
cat > $ISODIR/boot/grub/grub.cfg << EOF
set timeout=5
set default=0
```

```
menuentry "Boot $DISTRO_NAME Live" {  
    linux /casper/vmlinuz boot=casper quiet splash  
    initrd /casper/initrd  
}  
EOF
```

```
# Copy kernel and initrd  
sudo cp $CHROOTDIR/boot/vmlinuz-* $ISODIR/casper/vmlinuz  
sudo cp $CHROOTDIR/boot/initrd.img-* $ISODIR/casper/initrd
```

```
# Step 6: Create ISO image  
echo "[INFO] Creating ISO image..."  
grub-mkrescue -o $ISO_OUTPUT $ISODIR --compress=xz  
echo "[INFO] ISO image created: $ISO_OUTPUT"
```

```
# Clean up  
echo "[INFO] Cleaning up..."  
sudo umount $CHROOTDIR/{dev,proc,sys} | | true  
sudo rm -rf $WORKDIR
```

```
echo "[INFO] Build completed. ISO: $ISO_OUTPUT"
```