--docdir 입력하기 귀찮음

```
docdir () {
    local PATH_PREFIX_FOR_DOCDIR="/usr/share/doc"
    echo "${PATH_PREFIX_FOR_DOCDIR}/$(basename $(pwd))"
}

export -f docdir
```

Host에서 LFS 디렉토리로 chroot

6장 완료후 책의 끝까지 유용하게 사용하는 스크립트

사용법:
auto-lfs s : chroot 실행, 필요한 시스템 디렉토리 마운트
auto-lfs c : chroot 종료후 불필요한 시스템 디렉토리 마운트 해제

LFS, LFS_PART와 /boot, /boot/efi, /home, /sources 등은 상황에 맞게 조정이 필요합니다.

```
#!/bin/bash
# LFS 12.1-systemd에 맞춤
# <7.2 소유자 변경> 이후에 사용
# 커널 가상 파일 마운트/언마운트 지원용 스크립트
# Begin ~/bin/lfs-env-config

VERSION=0.1.7
LFS=/mnt/lfs
LFS_PART=/dev/nvme0n1p8
LFS_BOOT=/dev/nvme0n1p6
LFS_EFI=/dev/nvme0n1p2
LFS_HOME=/dev/nvme0n1p7

# 필요한 별도의 파티션 마운트
if ! mountpoint -q $LFS;          then mount $LFS_PART $LFS; fi
if ! mountpoint -q $LFS/boot;     then mount $LFS_BOOT $LFS/boot; fi
if ! mountpoint -q $LFS/boot/efi; then mount $LFS_EFI $LFS/boot/efi; fi
if ! mountpoint -q $LFS/home;     then mount $LFS_HOME $LFS/home; fi
if ! mountpoint -q $LFS/sources;  then mount /mnt/nfs/sources $LFS/sources;
fi

# /dev 존재 확인 후 필요한 조치 진행
if [ ! -d $LFS/dev ]; then
        echo "Make necessary directories"
        mkdir -pv $LFS/{dev,proc,sys,run}
```

```
else
        echo "All required directories have been verified."
fi

# 입력에 따라 chroot 실행 또는 exit 후 사용한 커널 가상 파일 시스템 언마운트
case $1 in
    clean|c|-c)
        if mountpoint -q $LFS/dev/shm; then
            # Unmount Kernel Virtual File System used in previous LFS
sessions.
            echo "LFS is now set to '"$LFS"'"
            echo "Now proceed with the unmount of <Kernel Virtual File
System> for chroot"
            for kvfs in dev/shm dev/pts sys proc run dev ; do
                    if mountpoint $LFS/$kvfs; then umount -q $LFS/$kvfs; fi
            done
            findmnt -D -R $LFS
            echo "Ok unmount unnecessary devices"
            echo "Good bye"
            exit 0
        else
            echo "Cleanup of Kernel Virtual File System is Complete."
            findmnt -D -R $LFS
            exit 0
        fi
    ;;
    start|s|-s)
        # Mount the Kernel Virtual File System to Enter LFS chroot
environment.
        echo "Now proceed with the mount of <Kernel Virtual File System> for
chroot"
        echo "LFS is now set to '"$LFS"'"
        if ! mountpoint -q $LFS/dev;     then mount --bind /dev $LFS/dev;
fi
        if ! mountpoint -q $LFS/dev/pts; then mount -t devpts devpts -o
gid=5,mode=0620 $LFS/dev/pts; fi
        if ! mountpoint -q $LFS/proc;    then mount -t proc proc $LFS/proc;
fi
        if ! mountpoint -q $LFS/sys;     then mount -t sysfs sysfs $LFS/sys;
fi
        if ! mountpoint -q $LFS/run;     then mount -t tmpfs tmpfs $LFS/run;
fi
        if ! mountpoint -q $LFS/dev/shm; then mount -t tmpfs -o nosuid,nodev
tmpfs $LFS/dev/shm;      fi
        if ! mountpoint -q $LFS/sources; then mount --rbind /mnt/nfs/sources
$LFS/sources;            fi
        findmnt -D -R $LFS
        # Run 'chroot'
        chroot "$LFS" $(type -p env) -i \
            HOME=/root                  \
            TERM="$TERM"                \
```

```
            PS1='<LFS> \u@\h [ \w ] \$ '\
            PATH=/usr/bin:/usr/sbin     \
            MAKEFLAGS="-j$(nproc)"      \
            TESTSUITEFLAGS="-j$(nproc)" \
            NINJAJOBS="$(nproc)"        \
            /bin/bash --login
        exit 0
    ;;
    version|v|-v)
        echo "LFS chroot script $VERSION"
        exit 0
    ;;
    help|h|-h)
        echo "LFS chroot script $VERSION"
        echo "Usage:"
        echo "clean   or c- Unmount Kernel Virtual File System"
        echo "start   or s- Mount and run 'chroot' command with necessary
variables"
        echo "version or v- Print script version"
        echo "help    or h - This screen"
        exit 0
    ;;
    *)
        #Guidance on the factors needed to execute script
        echo "sh ./lfs-env-config [ start | clean | help | version ]"
        exit 0
    ;;
esac
## End ~/bin/lfs-env-config
```

파일 다운로드 링크 처리

다음과 같이 등록하고 사용해야 cd 가 작동을 합니다.

```
alias <filename>='source <filename>'
```

다운로드 링크 입력: 다운로드 → 압축 해제 → 디렉토리 이동

추가한 사항

1. 입력받은 URL이 Git인 경우 처리 추가
2. 확장자가 .zip인 경우 처리 추가
3. 이미 다운로드 완료한 파일의 경우 처리 추가
4. 작업 디렉토리의 소유권 확인 추가
5. 출력 색상 변수 설정

추가할 사항

1. alias 등록 확인
2. wget, git, gawk 설치 여부 확인

수정할 사항

1. git clone 사용 할 때 작업 디렉터리에 먼저 받고 source 디렉터리에 복사하는 방식으로 변경 필요.

작성

```
user@host:[ ~ ] $ pre-stage
https://xorg.freedesktop.org/archive/individual/proto/xcb-proto-1.17.0.tar.x
z
xcb-proto-1.17.0.tar 100%
[=======================================================>]  148.19K  302.76KB/s
                          [Files: 1  Bytes: 148.19K [153.40KB/s] Redirects:
0  Todo: 0  Errors: 0    ]
user@host:[ /workbench/xcb-proto-1.17.0 ] $ _
```

```
#!/bin/bash

Green='\033[0;32m'
BIGreen='\033[1;32m'
Red='\033[0;31m'
Yellow='\033[0;33m'
Color_Off='\033[0m'

if [ "$(type -t p4c)" != 'alias' ] ; then
        echo -e "p4c is not ${Yellow}alias registered${Color_Off}, so the
functionality is not available."
        source ${HOME}/.bash_aliases
        return 2>&/dev/null
fi

if [ -z "$1" ]; then echo -e "${Green}An appropriate factor (file name or
download link) is required.${Color_Off}"; return; fi
InputStringType=$(echo "$1" | awk 'BEGIN { FS = ":" } ; { print $1 }')
BuildBase="$LFS"/workbench
SourceBase="$LFS"/sources
TargetFile=$(basename "$1")
GitDirCheck=$(basename -s .git "$1")
GitCheck=${TargetFile: -3}
ZipCheck=${TargetFile: -3}
TargetDir=$(echo "$TargetFile" | sed -e 's/\.tar.*$//' -e 's/\.tgz*$//' -e
's/\.src*$//' -e 's/\.zip*$//' -e 's@^.*/@@')
NotFoundMsg1="The directory derived from the filename is
${BIGreen}$TargetDir${Color_Off}.\n\
However, it does not exist, so it will move to the ${Yellow}most recent
directory created${Color_Off}.\n\
Please check if it is correct."
```

```
if [ $(stat -c %u $BuildBase) -ne $(id -u) ] ; then echo -e "Check
${Red}$BuildBase${Color_Off} Ownership"; return; fi

function chg_dir() {
    if [ -d "$1" ]; then
        cd "$1"
        else
                cd $(ls -tcA -w1 | head -n1)
                echo -e "$NotFoundMsg1"
        fi
}

# Change directory "$BuildBase"
if [ "$PWD" != "$BuildBase" ]; then
        chg_dir "$BuildBase"
fi

## Handling Git
if [ "$GitCheck" == "git" ] ; then
        if [ -d "$GitdirCheck" ] ; then
        chg_dir "$GitDirCheck"
                return
        else
            pushd $SourceBase
            if [ -d $GitDirCheck ] ; then
                rm -rf $GitDirCheck
            fi
            git clone "$1"
            cp -R $GitDirCheck $BuildBase
            popd
#        echo "Git #1"
        chg_dir "$GitDirCheck"
        return
    fi
fi

case "$InputStringType" in
        https|http|ftp)
            wget --no-verbose -nc --directory-prefix="$SourceBase" "$1"
        ;;
        *)
        ;;
esac

# Prepare Package Build
if [ "$ZipCheck" == "zip" ] ; then
        mkdir -p "$TargetDir"
        chg_dir "$TargetDir"
        unzip -q "$SourceBase"/"$TargetFile"
        return
else
```

```
        if [ ! -d "$TargetDir" ]; then
                tar -xf "$SourceBase"/"$TargetFile" 2>/dev/null
        fi
        chg_dir "$TargetDir"

fi

unset  InputStringType BuildBase SourceBase TargetFile GitDirCheck GitCheck
ZipCheck TargetDir NotFoundMsg1
```

구글 AI 모드로 정리: 미확인 상태

```
# 쉬뱅(#! /bin/bash)은 source용이라면 생략 가능

# 1. tput을 이용한 색상 정의
BOLD=$(tput bold)
BLUE=$(tput setaf 4)
GREEN=$(tput setaf 2)
BIGREEN="${BOLD}${GREEN}"
RED=$(tput setaf 1)
YELLOW=$(tput setaf 3)
RESET=$(tput sgr0)

# OpenRC 스타일 상태 출력 함수
print_status() {
    local msg="$1"
    local status=$2
    local col=$(tput cols)
    printf " ${BLUE}${BOLD}*${RESET} %s" "$msg"
    tput hpa $((col - 7))
    [ $status -eq 0 ] && printf "[  ${GREEN}OK${RESET}  ]\n" || printf "[
${RED}!!${RESET}  ]\n"
}

# 2. 초기 체크 (p4c 별칭 확인)
if [ "$(type -t p4c)" != 'alias' ]; then
    echo -e "p4c is not ${YELLOW}alias registered${RESET}, so functionality
is unavailable."
    # source 시에만 동작하도록
    [ -f "${HOME}/.bash_aliases" ] && . "${HOME}/.bash_aliases"
    return 2 2>/dev/null
fi

# 3. 변수 설정
if [ -z "$1" ]; then
    echo -e "${GREEN}An appropriate factor (file name or download link) is
required.${RESET}"
    return
fi
```

```
# 기본 경로 설정 (LFS 환경 변수 활용)
BuildBase="${LFS}/workbench"
SourceBase="${LFS}/sources"
TargetFile=$(basename "$1")
TargetDir=$(echo "$TargetFile" | sed -r
's/\.tar(\..*)?$|\.tgz$|\.src$|\.zip$//;s/\.git$//')
InputScheme=$(echo "$1" | grep -oP '^[^:]+(?=:)') # http, https, ftp 추출

# 4. 권한 체크
if [ ! -w "$BuildBase" ]; then
    echo -e "Check ${RED}$BuildBase${RESET} Ownership/Write Permission"
    return
fi

# 5. 디렉토리 이동 함수 (가장 최근 디렉토리 이동 로직 포함)
chg_dir() {
    if [ -d "$1" ]; then
        cd "$1"
    else
        local latest=$(ls -tcA -w1 | head -n1)
        cd "$latest"
        echo -e "Derived: ${BIGREEN}$1${RESET} (Not found)"
        echo -e "Moved to latest: ${YELLOW}$latest${RESET}"
    fi
}

# 작업 시작
[ "$PWD" != "$BuildBase" ] && cd "$BuildBase"

# 6. 원격 파일 다운로드 (http, https, ftp)
case "$InputScheme" in
    http|https|ftp)
        wget --no-verbose -nc -P "$SourceBase" "$1"
        print_status "Downloading $TargetFile" $?
        ;;
esac

# 7. 소스 처리 (Git vs Archive)
if [[ "$1" == *.git ]] || [ "$InputScheme" == "git" ]; then
    # Git 처리
    GitName=$(basename -s .git "$1")
    if [ -d "$GitName" ]; then
        chg_dir "$GitName"
    else
        pushd "$SourceBase" >/dev/null
        [ -d "$GitName" ] && rm -rf "$GitName"
        git clone "$1" && cp -R "$GitName" "$BuildBase/"
        popd >/dev/null
        chg_dir "$GitName"
        print_status "Git Clone: $GitName" $?
    fi
else
```

```
  # 압축 파일 처리
    if [[ "$TargetFile" == *.zip ]]; then
        mkdir -p "$TargetDir" && cd "$TargetDir"
        unzip -q "$SourceBase/$TargetFile"
        print_status "Unzip: $TargetFile" $?
    else
        if [ ! -d "$TargetDir" ]; then
            tar -xf "$SourceBase/$TargetFile" 2>/dev/null
            print_status "Extract: $TargetFile" $?
        fi
        chg_dir "$TargetDir"
    fi
fi

# 변수 정리 (local을 쓰지 못하는 source 환경이므로 직접 unset)
unset BOLD BLUE GREEN BIGREEN RED YELLOW RESET BuildBase SourceBase
TargetFile TargetDir InputScheme GitName
```