

# Hello world 에서 패키징까지

Kim Jinseok, Ubuntu Korea Community

# 들어가기 앞서

패키지란

의존성

소스와 바이너리 패키지

런치패드와 PGP key

## 패키지란 ?

- 배포를 위해 컴퓨터프로그램 뿐만아니라 메타데이터를 포함하고 있는 파일
- 메타데이터에는 패키지에 대한 설명 , 버전과 의존성 등을 담고있다 .

# 의존성

- 사전에 설치해야하는 다른 패키지
- 패키지의 빌드시 필요한 빌드 의존성도 존재

# 소스와 바이너리 패키지

## 소스 패키지

- 빌드 후 설치가 이루어지는 패키지
- 배포판, 아키텍처간 구분이 없음

## 바이너리 패키지

- 설치하려는 환경에 맞는 패키지를 설치
- 배포판과 아키텍처마다 따로 만들어짐

# 소스와 바이너리 패키지

배포판마다 다른 패키지 매니저와 포맷

-Ubuntu

apt

.deb

-RHEL

yum, dnf

.rpm

여러 배포판을 지원하는 패키지 매니저가 생기는 추세

-snappy

-flatpak

# 런치패드와 PGP Key

## 런치패드

- 개인과 팀이 소프트웨어 프로젝트에서 함께 작업할 수 있게 도와주는 오픈 소스 도구 모음

## PGP Key

- 강력한 비대칭 암호화 방식
- 인터넷에서 암호화를 하거나 전자서명을 하기 위해 사용됨
- 런치패드에서는 PC에서 한 작업에 서명하는 데 사용함

# 데비안 패키징

## upsteam tarball

- upstream 개발자가 작성한 소프트웨어 패키지를 말하며 tarball 이라고도 함

## Tarball

- .tar.gz .tgz .tar.bz2 .tar.xz

- tarball 은 데비안이 취한 것과 일치하며 패키지를 빌드 함



# 데비안 패키징

Debian source package 는 3 개의 파일

- 이름이 패턴에 맞게 수정된 tarball
- 데비안 패키지를 위해 추가된 파일들이 있는 debian directory  
debain.tar.gz(xz) 로 끝남
- 다른 2 개의 파일의 리스트가 있는 description 파일 (.dsc)

Debian binary package

- Source package 에서 빌드된 패키지
- 패키지 매니저로 설치 가능

# 데비안 패키징

데비안 패키지 버전

Upstream version-Debian version

Ex:1.0-1

1.0 이 upstream version -1 이 데비안 버전

우분투 패키지 버전

Upstream version-Debian version-Ubuntu version

Ex:1.0-1ubuntu1

Debian version 1.0-1 을 기반으로한 우분투의 1<sup>st</sup> 버전

Ex:1.0ubuntu1

Upsteam 1.0 을 기반으로 한 우분투의 1<sup>st</sup> 버전

# 데비안 패키징

## Debian 폴더

### 필수 파일

-control

-changelog

-rules

-copyright

### 선택 파일

-compat

-package.dirs

## 데비안 패키징

debian/control

Apt, dpkg, aptitude 등의 패키지 매니저가 사용할 정보들이 있는 파일 Source Block 과 Package Block 이 소스 패키지와 바이너리 패키지의 정보를 담고 있다 .

# 데비안 패키징

debian/control

## Source Block

- Source 소스 패키지 이름
- Maintainer 패키지 관리자
- Priority 패키지의 중요도
- Build-dependency 빌드시 필요한 의존성

# 데비안 패키징

debian/control

Package block

-Package 패키지 이름

-Architecture 빌드할 아키텍처

any: 여러개의 바이너리 패키지를 만든다 .

all: 하나의 바이너리 패키지만 만든다 .

-Dependency 의존성

`${mics:Depends}`, `${shlibs:Depends}`

자동으로 의존성을 잡아주는 구문 그러나 source block 에서는 못씀

# 데비안 패키징

debian/changelog 구조  
패키지의 변경사항들을 담은 파일  
특정한 구조를 지니고 있음

```
name (version) distribution(s); urgency=[low, medium, high, emergency]
```

```
Message
```

```
-- Maintainer <e-mail> when timezone
```

## 데비안 패키징

debian/rules

dpkg-buildpackage 가 이 파일 이용하여 패키지를 만듦

dh\_ 로 시작하는 명령어들이 묶여 실행됨

Makefile 과 같은 문법을 사용한다 .



## 데비안 패키징

### Copyright

해당 패키지의 저작권을 명시하는 파일

*/usr/share/common-licenses* 에 *GPL BSD* 등의 라이선스들이 있다.

# PPA

PPA 를 사용하면 런치패드에서 apt 저장소로 빌드하고 게시할 우분투 소스 패키지를 업로드 할 수 있음

사용자들에게 직접 업데이트를 제공할 수 있음

Source package 를 런치패드 서버로 업로드 뒤 패키지의 BuildSocre 에 따라 큐가 정해짐  
빌드 팜에서 빌드 후 배포가 완료된다 .

# PPA

## Target Pocket

특정 배포판 시리즈와 관련된 패키지 그룹  
changelog 에서 distribution 에 따라 달라진다 .

-backports : 0  
<release>: 1500  
-proposed: 3000  
-updates:3000  
-security:4500

# PPA

## Target Components

타입 라이선스또는 레벨 따라 점수가 정해짐  
debian/copyright 로 명시한 라이선스를 기준

Multiverse : 0

Universe : 250

Restricted : 750

Main : 1000

Partiner : 1250

# PPA

## Urgency

debian/changelog 에 명시 되어있는 urgency

Low : 5

Medium : 10

Hight : 15

Emergency : 20

# 데비안 패키징

데비안 패키징을 할 **tarball**

helloworld-1.0.tar.gz  
helloworld-1.0/helloworld.c  
helloworld-1.0Makefile

```
prefix = /usr/local  
bindir = $(prefix)/bin
```

```
CFLAGS = --std=c99
```

```
all: helloworld
```

```
clean:  
    rm -f helloworld
```

```
install: all  
    install helloworld $(DESTDIR)$(bindir)
```

# 데비안 패키징

## 필요 패키지 설치

```
sudo apt install build-essential devscripts debhelper
```

Build-essential

데비안 패키지를 빌드할 수 있도록 환경구축

Devscript

패키지 유지보수를 쉽게 만들어주는 패키지

Debhelper

debian/rules 스크립트의 helper

# 데비안 패키징

일반 소스 패키지의 이름을 데비안 패키징 패턴에 맞게 변경

P\_V.orig.tar.gz

```
mv helloworld-1.0.tar.gz helloworld_1.0.orig.tar.gz
```

Upstream tarball 버전에서는 dash(-) 대신 underscore(\_)를 사용



# 데비안 패키징

## 데비안 패키징 파일 생성 - 1

```
tar xf helloworld_1.0.orig.tar.gz  
cd hellowolrd-1.0
```

모든 파일들은 이 폴더 내에서 생성, 수정이 이루어집니다.  
(폴더 밖에서 수정이 이루어지면 에러를 일으킵니다.)

```
mkdir debian
```

## 데비안 패키징

패키지의 변경사항을 담은 changelog 파일을 만듭니다.

```
dch --create -v 1.0-1 --package helloworld
```

# 데비안 패키징

## 기본 폼

```
helloworld (1.0-1) UNRELEASED; urgency=medium
```

```
* Initial release. (Closes: #XXXXXX)
```

```
-- fmowl <fmowl@fmowlpc> Tue, 06 Feb 2018 08:50:44 +0900
```

## 수정

```
helloworld (1.0-1) artful; urgency=medium
```

```
* Initial release.
```

```
-- KimJinseok <jinseok1001@hotmail.com> Tue, 06 Feb 2018 08:50:44 +0900
```

# 데비안 패키징

debian/compat 압축도 ( 새로운 패키지들은 10 으로 해놓음 )

```
10
```

debian/copyright 저작권 ( 빈 파일 )

debian/rules 빌드 레시피

```
#!/usr/bin/make -f
```

```
%:
```

```
dh @
```

# 데비안 패키징

## debian/control

```
Source: helloworld  
Maintainer: KimJinseok <jinseok1001@hotmail.com>  
Section: misc  
Priority: optional  
Build-Depends: debhelper (>= 10)  
Standards-Version: 4.1.1
```

```
Package: helloworld  
Priority: optional  
Architecture: any  
Description: hello world example  
Depends: ${mics:Depends}, ${shlibs:Depends}
```

# 데비안 패키징

## 첫 번째 빌드

```
debuild -us -uc
```

## 실패

```
LD AM_UPDATE_INFO_DIR=no
make[1]: Entering directory '/home/fmowl/build/helloworld-1.0'
install helloworld /home/fmowl/build/helloworld-1.0/debian/helloworld/usr/local/
bin
install: cannot create regular file '/home/fmowl/build/helloworld-1.0/debian/hel
loworld/usr/local/bin': No such file or directory
Makefile:12: recipe for target 'install' failed
make[1]: *** [install] Error 1
make[1]: Leaving directory '/home/fmowl/build/helloworld-1.0'
dh_auto_install: make -j8 install DESTDIR=/home/fmowl/build/helloworld-1.0/debia
n/helloworld AM_UPDATE_INFO_DIR=no returned exit code 2
debian/rules:3: recipe for target 'binary' failed
make: *** [binary] Error 2
dpkg-buildpackage: error: fakeroot debian/rules binary gave error exit status 2
debuild: fatal error at line 1124:
dpkg-buildpackage -rfakeroot -us -uc failed
```

# 데비안 패키징

## 실패 이유

- debian/helloworld/usr/bin 에 설치가 되어야 하는데 Makefile 에서 /usr/bin 에 설치하라고 명시를 해놓았기 때문
- debian/helloworld/usr/bin 이라는 디렉토리가 없음

debian/rules 에 설치 위치를 덮어쓰고  
package.dir 을 만들어 해당 디렉토리를 만들게 함

# 데비안 패키징

파일 수정 및 추가

debain/rules

```
#!/usr/bin/make -f
%:
    dh $@
override_dh_auto_install:
    $(MAKE) DESTDIR=$(pwd)/debian/helloworld prefix=/usr install
```

debian/helloworld.dir

usr/bin



# 데비안 패키징

두번째 빌드

```
debuild -us -uc
```

성공

```
ts  
helloworld-1.0                helloworld_1.0-1.debian.tar.xz  
helloworld_1.0-1_amd64.build  helloworld_1.0-1.dsc  
helloworld_1.0-1_amd64.buildinfo helloworld_1.0.orig.tar.gz  
helloworld_1.0-1_amd64.changes helloworld-dbgsym_1.0-1_amd64.ddeb  
helloworld_1.0-1_amd64.deb
```

# PPA

ppa 활성화 (launchpad.net 로그인 후 프로필 )

## Personal package archives

 helloworld

[+ Create a new PPA](#) [i View your private PPA subscriptions](#)

### URL:

http://ppa.launchpad.net/fmowl/

At least one lowercase letter or number, followed by letters, numbers, dots, hyphens or pluses.  
Keep this name short; it is used in URLs.

### Display name:

A short title for the archive.

or [Cancel](#)

# PPA

소스 패키지 빌드

```
debuild -S
```

소스 패키지 업로드

```
dput ppa:fmowl/helloworld helloworld_1.0-1_source.changes
```

```
dput ppa P_V_source.changes
```

업로드가 성공적이라면 빌드가 시작됨

# PPA

빌드 후 배포가 완료가 되면 해당 패키지를 설치 할 수 있게 됩니다 .

```
sudo apt-add-repository ppa:fmowl/helloworld  
sudo apt update  
sudo apt install helloworld  
helloworld  
Hello, World!
```

Q&A