

Introduction to Hanjp-IM Project

An IME for typing Japanese using Korean Hangul

Youngbin Han

Ubuntu Korea Community
(Ubuntu Korean LoCo Team)
<sukso96100@gmail.com>

DebConf18 – Hsinchu, Taiwan

\$ whoami

- Current Leader of Ubuntu Korea Community(Ubuntu Korean LoCo Team)
- Undergraduate student majoring Software Engineering
 - and professors teaches us Web Development(Java, Spring)
- Interested in Web, Android, Open Source, Linux Desktop, Input Method.

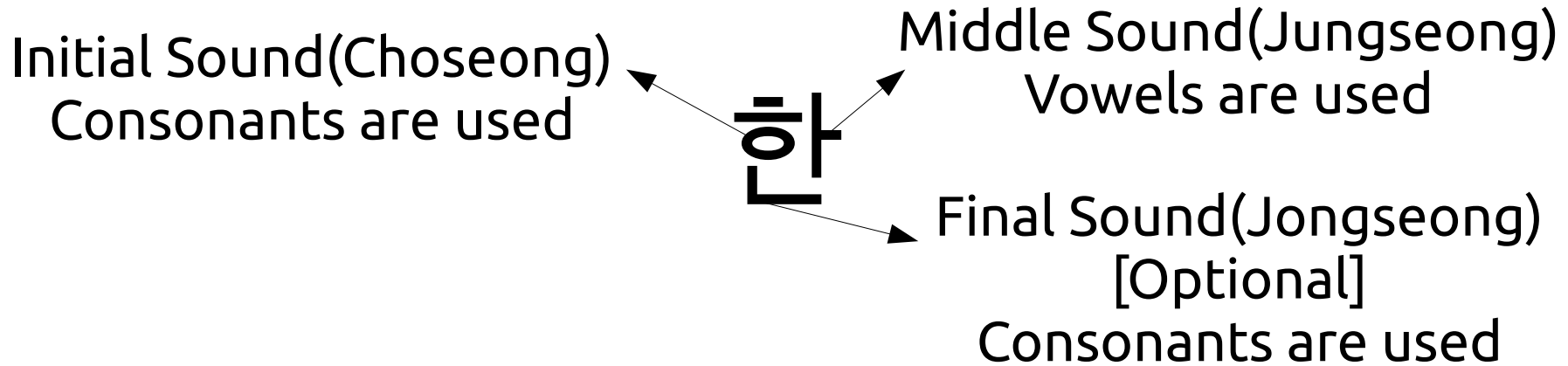
What is this project?

- A project for making another Japanese Input Method.
- An IM that can type Japanese using Korean Hangul.

Type Japanese using Hangul?

- It's like typing Japanese with romaji IM but it uses Hangul(Korean characters) instead of Latin(or Roman) script.
- Type pronunciation of Japanese characters(Hiragana, Katakana, Kanji...) in Korean Hangul then the IM converts it automatically into Japanese characters.

Hangul – Korean Script



ㅎ ㅎ ㅏ ㅏ ㄴ ㄴ 하

Basic idea

Typing “Kon ni chi wa”(means “Hello”)

underlined : preedit / other : committed string

- Romaji Input

- ko
- こn
- こんni
- こんにchi
- こんにちwa
- こんにちわ

- Hanjip Input Scenario

- 코
- こㄴ
- こんㄴ
- こんに치
- こんにち와
- こんにちわ

Purpose?

- To provide an easy way for Korean to type Japanese who just get started to learn Japanese.
- Just for fun(?)
- To learn and experience how open source and collaboration works.
- (For my cases,) To learn how Input Method Works.

- Project was proposed by Gwangyeon Kim at last September.
- I contacted him to participate and help him gathering members.
- We started the project at last December.



Gwangyeon Kim
(Project Leader)



Hoseok Shin


Juneyoung Kang



Youngbin Han
(Me)



Original
Members
(at the
beginning)



Gwangyeon Kim
(Project Leader)

Hoseok Shin
Retired for his job

Youngbin Han
(Me)

Juneyoung Kang
Retired temporary
for startup

Current
Members

Original plan

- Use uim for building IM.
 - Uim uses scheme for writing IM.
 - We expected that it would be similar as writing things in python.
 - Referenced uim-byeoru(Korean IM for uim)'s code a lot.
- Match each Hangul syllables with Kana.
- Introduce some of Yethangul(Old Hangul) to type pronunciation that can't be typed with just Modern Hangul.

Keyboard Layout

~ 、	! 1	@ 2	# 3	\$ 4	% 5	^ 6	& 7	* 8	(9) 0	- _	+ =	 ¥	←
Tab ↔	Q ㅅ ㅅ	W ㅅ ㅅ	E ㅅ ㅅ	R ㅅ ㅅ	T ㅅ ㅅ	Y ㅅ ㅅ	U ㅅ ㅅ	I ㅅ ㅅ	O ㅅ ㅅ	P ㅅ ㅅ	{ [}]	
Caps Lock 히라가나/가타카나	A ㅅ ㅅ	S ㅅ ㅅ	D ㅅ ㅅ	F ㅅ ㅅ	G ㅅ ㅅ	H ㅅ ㅅ	J ㅅ ㅅ	K ㅅ ㅅ	L ㅅ ㅅ	: ;	" '		↵ Enter	
Shift ⬆	Z ㅅ ㅅ	X ㅅ ㅅ	C ㅅ ㅅ	V ㅅ ㅅ	B ㅅ ㅅ	N ㅅ ㅅ	M ㅅ ㅅ	< , 、	> . , 。	? /	Shift ⬆			
Ctrl	Alt	무변환	변환								일/영	Alt	Ctrl	

Yethangul(Old Hangul): ㅇ, ㅅ, ㅅ, ㅅ / Caps Lock: Hiragana, Katakana toggle

Reality

- Unlike languages like C, Python, Java, Scheme's grammar was very unfamiliar for us.
- Uim didn't have enough documents(API references, Example Projects, etc...), We had to analyze source of other IMs for uim.
- One-to-one Matching between Hangul syllable and Kana was inefficient.
- We had very few backgrounds for Input Methods
 - eg. Automata

One-to-one match

카[ka]	(((hj-choseong-k hj-jungseong-a). ()) ("か" "カ" "カ"))
키[ki]	(((hj-choseong-k hj-jungseong-i). ()) ("き" "キ" "キ"))
쿠(크)[ku]	(((hj-choseong-k hj-jungseong-u). ()) ("く" "ク" "ク"))
케(캐)[ke]	(((hj-choseong-k hj-jungseong-e). ()) ("け" "ケ" "ケ"))
코[ko]	(((hj-choseong-k hj-jungseong-o). ()) ("こ" "コ" "コ"))
가[ga]	(((hj-choseong-g hj-jungseong-a). ()) ("が" "ガ" "ガ"))
기[gi]	(((hj-choseong-g hj-jungseong-i). ()) ("ぎ" "ギ" "ギ"))
구(그)[gu]	(((hj-choseong-g hj-jungseong-u). ()) ("ぐ" "グ" "グ"))
게(개)[ge]	(((hj-choseong-g hj-jungseong-e). ()) ("げ" "ゲ" "ゲ"))
고[go]	(((hj-choseong-g hj-jungseong-o). ()) ("ご" "ゴ" "ゴ"))

And... bunch of more matches like this. Take a look at
<https://github.com/ubuntu-kr/hanj-p-im/blob/master/scm/japanese-hj.scm>

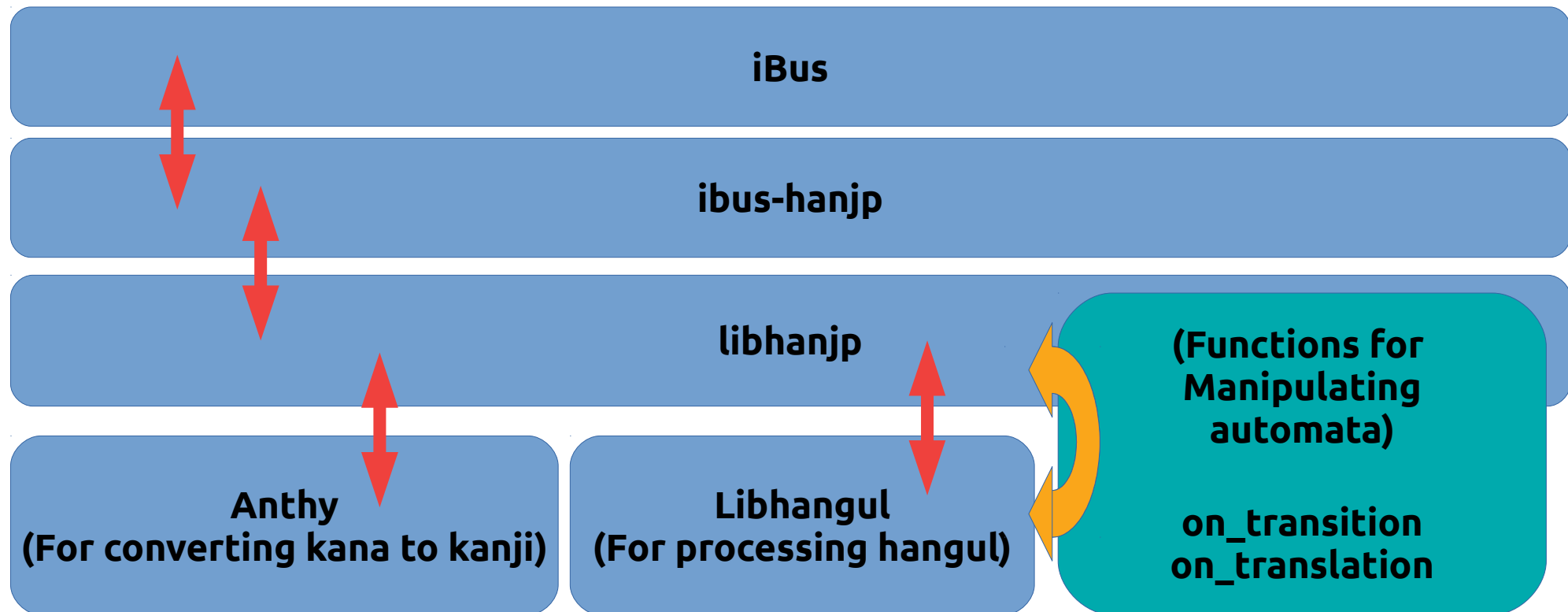
New approach

- Switch to iBus from Uim.
 - iBus has various resources for developers.
 - eg. Docs, API References, Example(Template) Projects
 - <https://github.com/phuang/ibus-tmpl>
 - <http://ibus.github.io/docs/ibus-1.5/index.html>
 - We can write IME for iBus in C which is much more familiar than Scheme.
 - Since we can write in C, we can also use libhangul for processing Hangul.

New approach

- Build separate library, “libhanjp” for processing Hangul to Kana conversion and its own input context.
 - Just like libhangul, which is a library for processing Hangul and Hangul input context.

New approach, New Design

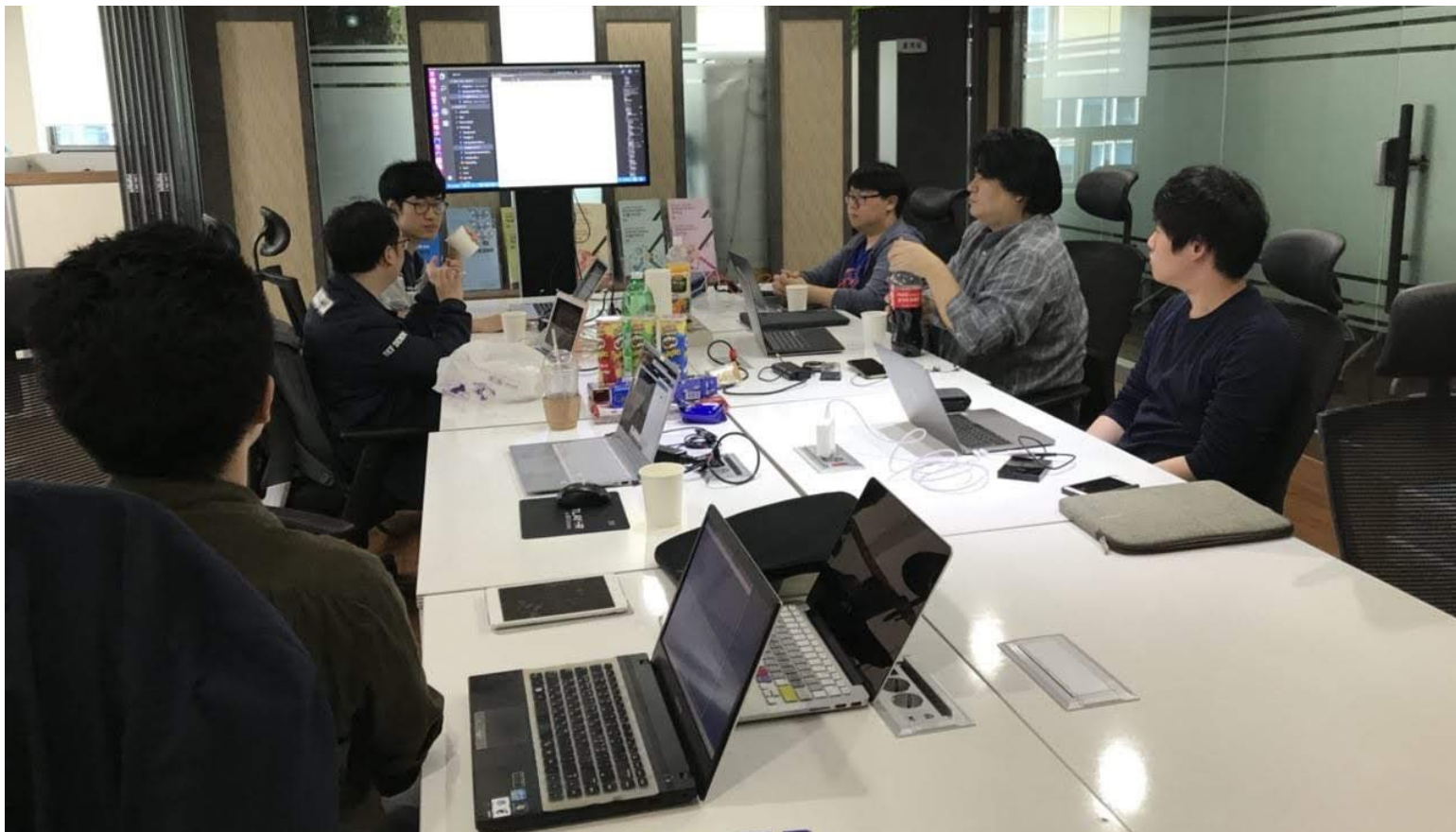


Presentation for midterm report

28 Apr. 2018



Joint meetup with GNOME Korea



Joint meetup with GNOME Korea

- Seong-Ho Cho from GNOME Korea Suggested a Joint meetup after watching the presentation.
- Discussed about Hangul Input and Hanjp-IM Project

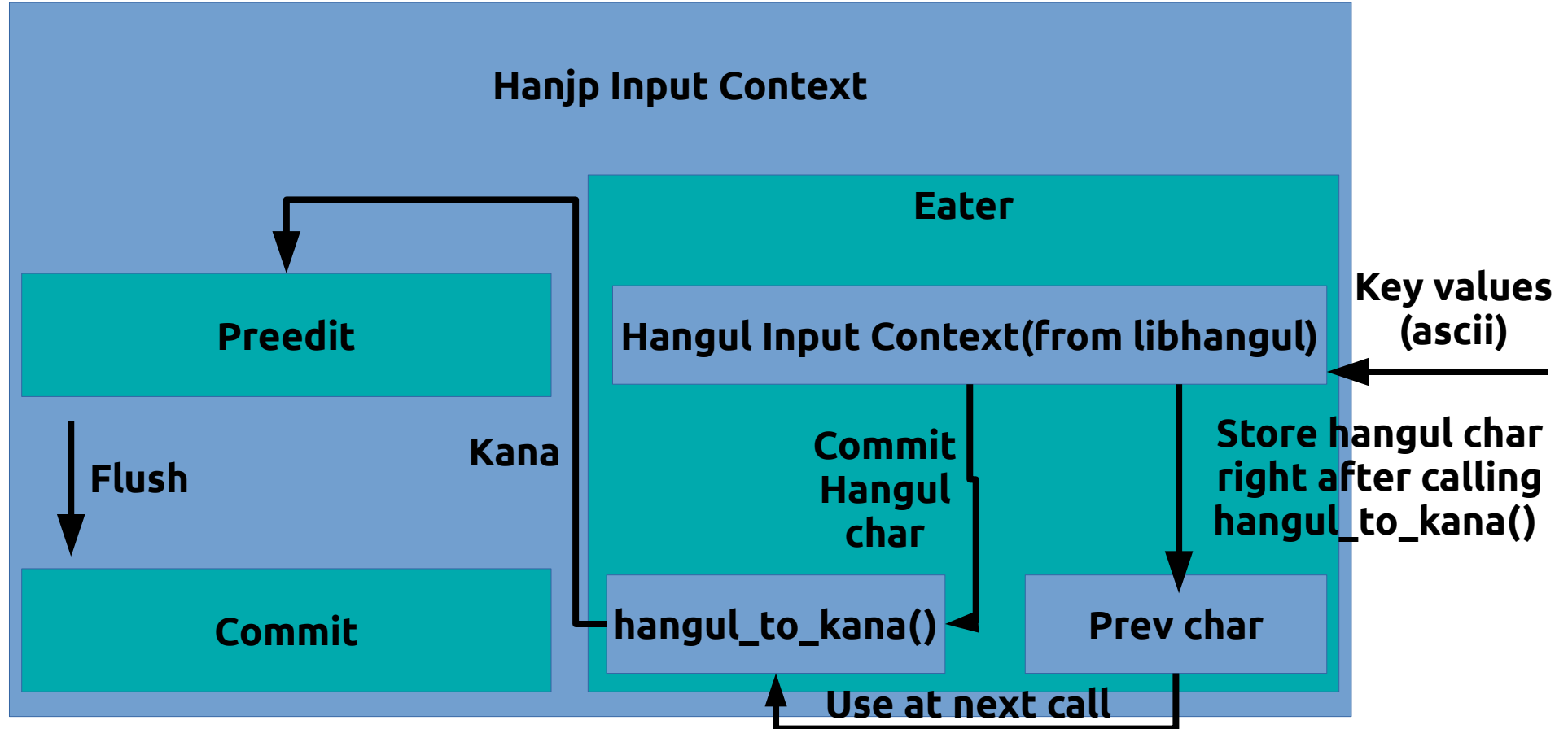
Joint meetup with GNOME Korea

- What we discussed and worked together on Hanjp-IM
 - Brief introduction about the project
 - libhanjp
 - Compiling – Working with dependencies, Makefile
 - Debugging with GDB
 - ibus-hanjp
 - Discussions about preedit and cursor handling

New approach failed again

- Gwangyeon Kim successfully compiled sample program for testing libhanjp, But not worked.
 - After debugging using with gdb with some help from Gwan-gyeong Mun from GNOME Korea, We found that libhanjp doesn't work properly.
 - It doesn't accepts hangul input properly and also not converts hangul to kana properly. In short, The whole libhanjp just doesn't work.
 - requires bunch of fix and time.

Another new approach



Converting Hangul to Kana

카

Committed Hangul
Char from HIC

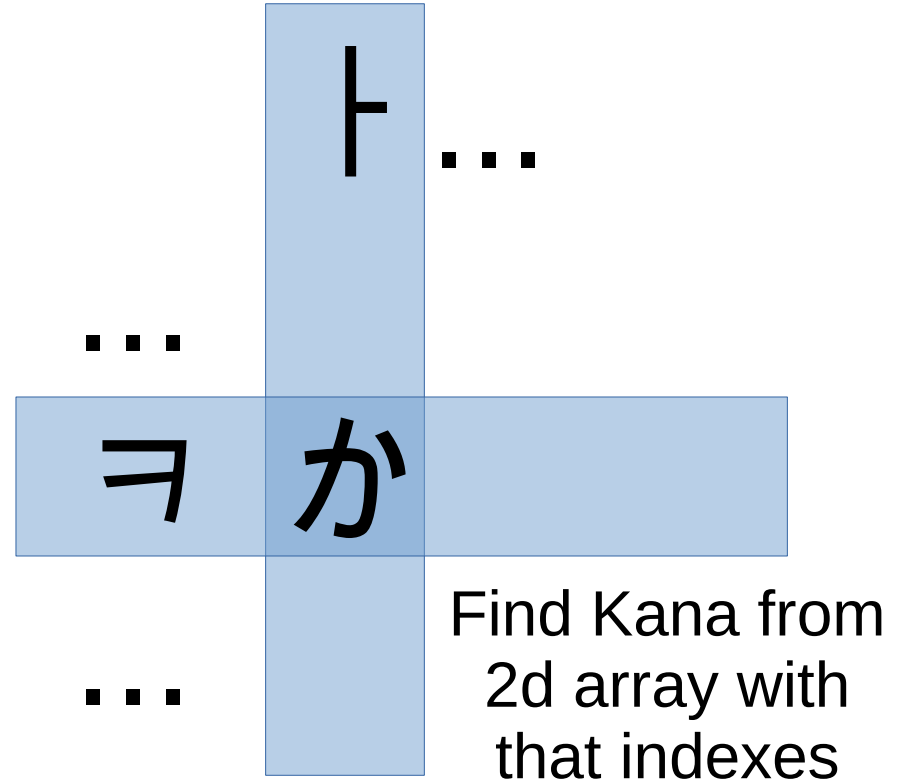
ㅋ : 1

ㅏ : 0

Find index for
each one

ㅋ, ㅏ

Ungroup the character



Current Status

- Implemented, But needs some tests.
 - hangul_to_kana function
 - Clear Sound(清音)
 - Voiced Sound(or Sonant, 濁音)
 - P-Sound(半濁音)

–

Current Status

- Discussions
 - Implementing Sokuon(っ/ツ, 促音) and Hatsuon(ん, 撥音) input.
 - Having a lot of discussions about this these days.
 - Current libhanjp architecture.
 - Needs some more discussions, it's not finalized.
 - Actually, the diagram at slide 23 was just updated a few days ago.(Yes, during DebConf18)
 - Since we haven't consider about items from To-Dos yet, It will definitely change

To-Dos

- Update ibus-hanjp for the latest libhanjp architecture.
- Implementing Chōonpu(Long sound) using :◌◌(Hangul double dot tone mark)
- Katakana Input
- Converting Kana to Kanji(Chinese characters) in preedit
 - Use Anthy for conversion
- **Build a Working Prototype IME!(We really want this!)**
 - We don't have any working prototype yet.
 - First goal of this is to make basic Kana Input work.
 - Making Kanji conversion work would be our next goal.

Getting Involved!

- All current members(including me) are Korean, and we are not professional(?), fluent or native Japanese user.
 - There can be some wrong implementations!
- Send your advices, suggestions, opinions with email or submit a new issue with it.

Getting Involved!

- Participations from non-Korean
 - We really welcome submitting issues. Please submit new one if you have any opinions.
 - Sorry, We're not accepting patches, pull requests or new core members from foreigners for now.
 - We got no working prototypes yet. And we really want to focus on building working prototype.
 - We might able to accept patches or PR once we got a working prototype.
 - Our team's activity is currently based on weekly offline meeting in Seoul. We need to replace this with something like online meetings in IRC, mailing list, etc. and we need to discuss about that.

Some Links

- Report for Joint meetup with GNOME Korea
 - <http://blog.ubuntu-kr.org/2018/05/19/hanjp-im-gnome-kr-joint-meetup-en.html>
- GitHub Repository
 - <https://github.com/ubuntu-kr/hanjp-im>
 - Includes codes we gave up working and codes we are currently working on.
 - Also includes documentations for hangul to kana conversion table and conversion rules.

ubuntu-kr

- Hanjp-IM Project is hosted by Ubuntu Korea Community(Ubuntu Korean LoCo Team)
- <https://www.ubuntu-kr.org>
- <https://wiki.ubuntu.com/KoreanTeam>
- korea.ubuntu@gmail.com

Questions?

- Ask now or ask at noisy hacklab.
- Or send it with email – sukso96100@gmail.com
 - No later than Sep. 6th please. I'm going to military service(Air Force) at Sep. 17th.
 - After Sep. 17th, Contact with Gwangyeon Kim the project leader - hemool123@naver.com

Thanks!