Google Summer of Code

Project Proposal 2021

Customize IBus for Separate Themes

Hollow Man

Last Modified Date: Monday, 12 April 2021

Contents

1. My Interest

Last year I actively proposed adding Table Input Feature for ibus-libpinyin for GSoC 2020, but due to limited slot and project priority, the idea was cancelled, and I failed in application. So this year I aim to do IBus again since I have prepared a lot in last year's application.

IBus stands for Intelligent Input Bus. It is a novel input framework for Linux OS. It offers full featured and user-friendly input method user interface.

Here is a wiki link about IBus 1.5 API: https://github.com/ibus/ibus/wiki

2. My Experience

I'm Songlin Jiang, a Computer Science senior undergraduate from Lanzhou University, China. It's the first time for me to participate in the Google Summer of Code activity.

Upon entering the university, I began to develop enthusiasm in Open Source guided by senior students at Lanzhou University Open Source Society. I think the most intriguing thing about open source is that people from various backgrounds work across different timelines without borders to achieve one common goal, the success of their projects without expecting anything in return. I set up my own GitHub account (@HollowMan6). Whenever I wrote some interesting programs, I would make a repository for it and post it on GitHub. I appreciate the open-source spirit and devote most of my free time to contributing to GitHub. I have contributed to many popular projects and 80 organizations including TensorFlow, Node.js, Python, Kubernetes, etc., which helps me gain over 1.4k followers and 205 personal project stars on GitHub. Moreover, Open Source has also blessed me with a lot of new knowledge and skills. By referring to the open-source project and tutorials on GitHub, I have learned by myself knowledge related to Go, React, Vue.js, C#, Flutter, Arduino, Android App Development that can't be found in the university undergraduate curriculum. These bits of knowledge have greatly broadened my vision.

A. Experience with the development tools

Javascript: I have extensive experience with Javascript including ES5, ES6, Node.JS. I have also done a lot of projects using Javascript, here's a list.

- **ChaoXing ErYa & Wisdom Tree Lessons Helper**

Developed by myself alone, it's a chrome plugin for automating course playing processes. So far it has got 20 stars and 4 forks.

Link: https://github.com/HollowMan6/ChaoXing-ErYa-Wisdom-Tree-Lessons-Helper

- **My Home Page https://hollowman.ml**

The index and funding pages are mainly coded by me along without using framework, and of course with a lot of hidden functionalities. (see https://github.com/HollowMan6/HollowMan6.github.io#function-surprise for more)

I have also created several JS repositories for the background canvas source code used in my Home Page.

Links: https://github.com/HollowMan6/HollowMan6.github.io

https://github.com/HollowMan6/Staggered-Mouse-Balls

https://github.com/HollowMan6/canvas-explosion-click.js

https://github.com/hollowman6/canvas-ribbon.js

https://github.com/hollowman6/canvas-ball.js

https://github.com/hollowman6/canvas-nest.js

https://github.com/hollowman6/sun.js

- **Emacs Application Framework**

I participated in this project as a student of Open Source Promotion Plan – Summer 2020 (an activity pretty much like GSoC held by Intelligent Software Research Centre, Institute of Software, Chinese Academy of Sciences in China). During the project I completed 15 compulsories + 3 voluntarily different level tasks using Emacs Lisp, Python, JavaScript, SQLite to add new features for Emacs Application Framework (EAF). One of my greatest contribution was that I added carrot browsing function using JavaScript for the browser in EAF that got the Owner and Project Founder's high appraisal (In Chinese): https://manateelazycat.github.io/emacs/eaf/2020/07/22/eaf-support-caret-browse.html

Here's the detailed tasks that I have done (In Chinese): https://github.com/HollowMan6/My-OSPP-Summer/blob/master/2020-Final-Report.md

Link to the project: https://github.com/manateelazycat/emacs-application-framework

Python: I also have extensive experience with Python, including a lot of web crawling projects, here's a list: https://github.com/users/HollowMan6/projects/1

- **openEuler University Developer Competition**

Here's the home page of the competition (in Chinese) https://www.oschina.net/2020-openeuler. I've participated in this activity and used python to realize a package update notifying software with dnf. Though it's a team-cooperated competition, but I do the whole project alone. My team number is 1798858424, team name is Hollow Man, and here's the rematch review opinion from the competition organizer: https://gitee.com/openeuler-competition/topics-2020/blob/master/ReviewOpinionForRematch/2020%20openEuler%E9%AB%98%E6%A0%A1%E5%BC%80%E5%8F%91%E8%80%85%E5%A4%A7%E8%B5%9B-%E8%AF%84%E5%88%86%E5%8F%8A%E8%AF%84%E5%AE%A1%E6%84%8F%E8%A7%81.xlsx , I got 19/80 in the final entering competition for the project, https://my.oschina.net/u/4469669/blog/5014377

B. Experience with teams, online developer communities and large code bases

- **Arthas**

I participated in this project as a student of Alibaba Summer of Code 2020 (an internationalized activity pretty much like GSoC but held by Alibaba). Arthas is an Alibaba Java diagnostic tool, which is the most popular project in star number of Alibaba Group. During the project I wrote Chinese and English online tutorial for Arthas using Katacoda and Vue.js. I also used Java to add command parameters of searching Class Loader by name instead of Hash Code.

Here's the full task list that I have done: https://github.com/HollowMan6/My-Alibaba-Summer-of-Code/blob/master/2020-Arthas/Alibaba%20Summer%20of%20Code%202020%20-%20Arthas%20Final%20Report.md

Link to the project: https://github.com/alibaba/arthas

I have already made some patches of English Grammar Errors in the UI last year for the ibus-libpinyin: https://github.com/libpinyin/ibus-libpinyin/pull/224

This year I have sent a PR: https://github.com/libpinyin/ibus-libpinyin/pull/312 which fixed deprecated glib function g_atexit().

And just as I mentioned above, when the 2020 summer holiday began, I took two Open Source Internships online simultaneously, one at Emacs (China) community under the fund of Open Source Promotion Plan – Summer 2020 by Institute of Software, Chinese Academy of Sciences, another at Arthas community under the fund of Alibaba Summer of Code. During the summertime, I was quickly transformed from a novice of the two communities into a senior contributor and completed the task excellently through detailed plans. I also actively participated in the community discussions, explored solutions to the requirements even beyond the compulsory tasks, responded promptly, and carried out the change requests proposed by the tutors. All the tutors from the two communities gave me high appraisals of what I have done during the internships. (One of the mentors and developers from EAF: https://isrc.iscas.ac.cn/gitlab/summer2020/students/proj-2012153/-/issues/25#note_175301) (In Chinese) Through the two internships, I got my team cooperation and communication skills practiced. Moreover, I also got tremendous experience and skills in participating and contributing to big projects. Finally I contributed 49 commits, 4515 lines of code contribution for EAF ranking 5: https://github.com/manateelazycat/emacs-application-framework/graphs/contributors. And for Arthas 80 commits, 26734 lines of code or documentation contribution ranking 2: https://github.com/alibaba/arthas/graphs/contributors.

3. Proposed summer project

A. Aim

Since the original skin function of IBus in different desktop environments is based on GNOME-shell and GTK, we need to inherit and customize it. so, the aim is to add an extension to GNOME so that users can freely choose the style they need for IBus.

B. Coding

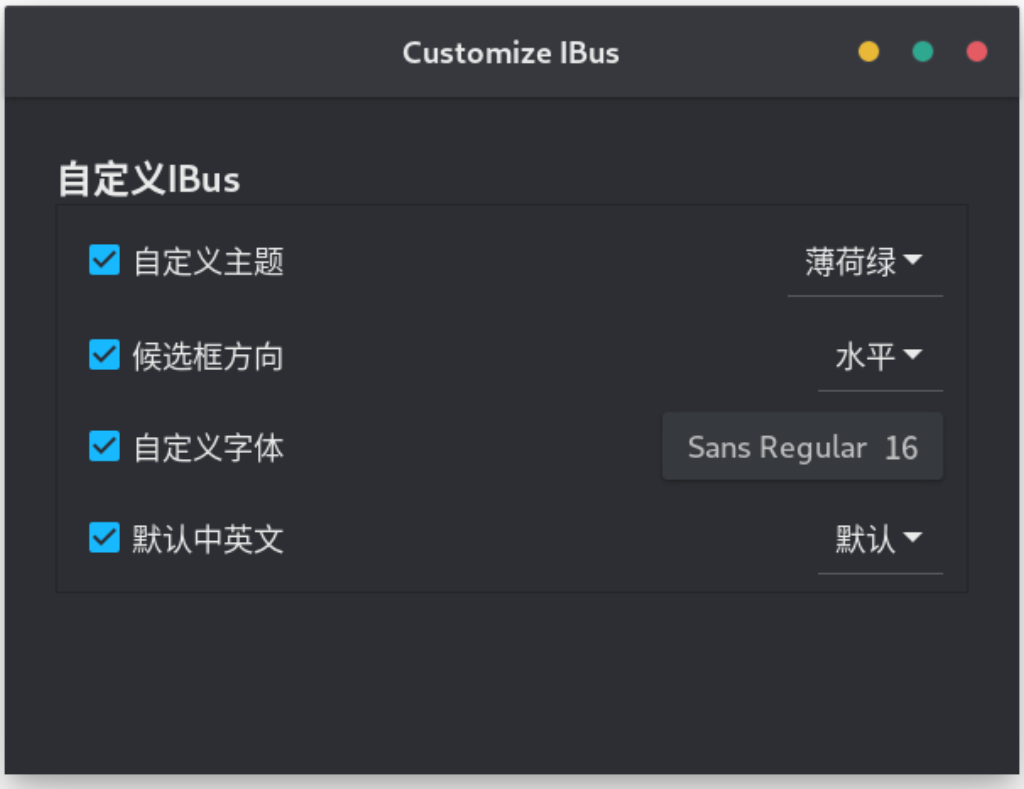
After searching on the Google, I noticed that currently ibus's theme is determined by the desktop theme https://github.com/rime/ibus-rime/issues/77, and we can't change it manually unless you use Gnome and change it with gnome-shell.

And another workaround to change the theme is to act in this way in Gnome: https://blog.csdn.net/weixin_30414245/article/details/96607094.

I've got some ideas from IBus Tweaker: https://extensions.gnome.org/extension/2820/ibus-tweaker/, https://github.com/tuberry/ibus-tweaker

We can make a GNOME Shell Extension that reads from the skin files (it can be rime-like color schemes https://github.com/rime-aca/color_schemes) located in the specified folder (e.g. ~/ibus_themes), users can choose a theme from the list or just follow the original GNOME theme.

I have already done some work to Import from ibus-tweaker and remove features unrelated to IBus: https://github.com/HollowMan6/Customize-IBus



After interacting with the mentors, using the current gtk3 theme and gnome-shell theme for this feature may be a good choice. We can refer to GNOME Shell Extension user theme to read from the local skin files: https://gitlab.gnome.org/GNOME/gnome-shell-extensions/-/tree/master/extensions/user-theme, I have managed to make a demo during application period.

Firstly, IBus relies on St.widget to set style for it. I have checked the GJS St.widget documentation https://gjs-docs.gnome.org/st10~1.0_api/st.widget, and unlike C St.widget https://developer.gnome.org/st/stable/StWidget.html, which provide a st_widget_set_theme() method and st_widget_get_theme() method, GJS doesn't provide such method for setting stylesheet themes for this class, but only get_theme_node(). And St.ThemeNode either doesn't provide any methods for getting a St.ThemeContext or a way to set stylesheet themes for it. I guess GNOME has intentionally disabled this feature for setting stylesheets for a single St.widget.

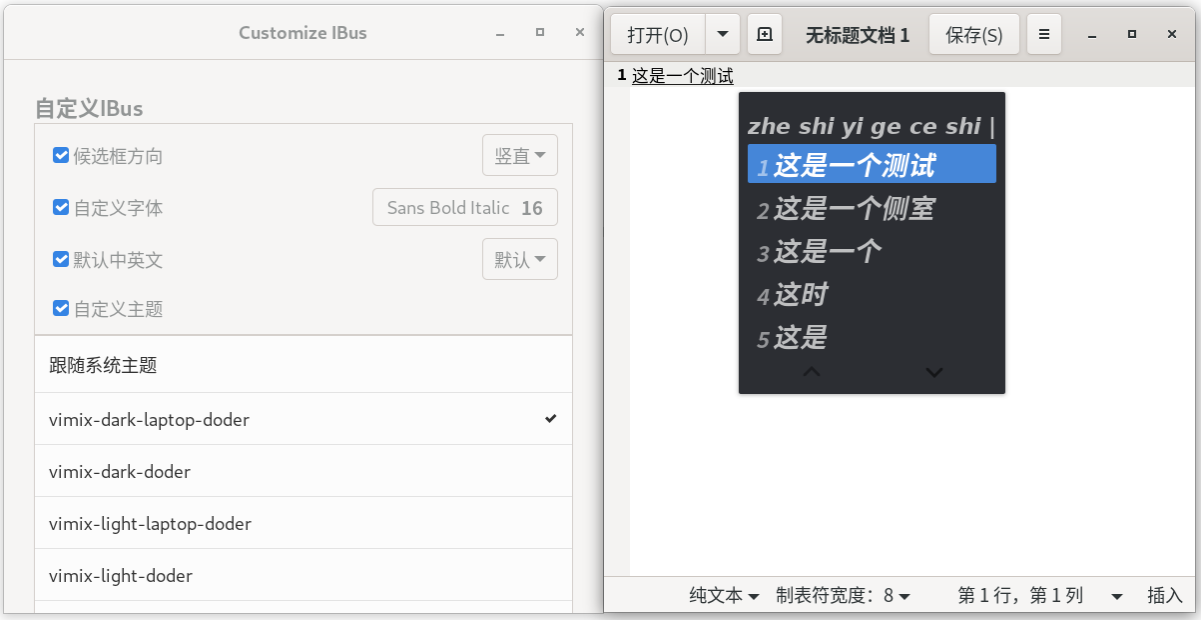
Then I tried importing those stylesheets into extension's stylesheet.css file, and it turned out that the whole systems' theme will change instead of only ibus. So I have no choice but to copy the related style class names in the user's stylesheet into extension's stylesheet.css.

When user chooses a theme from the list, this extension will first read the theme CSS file, extract the IBus related style classes (.candidate-*), then write it to extension's stylesheet.css. Finally restart the GNOME-shell.

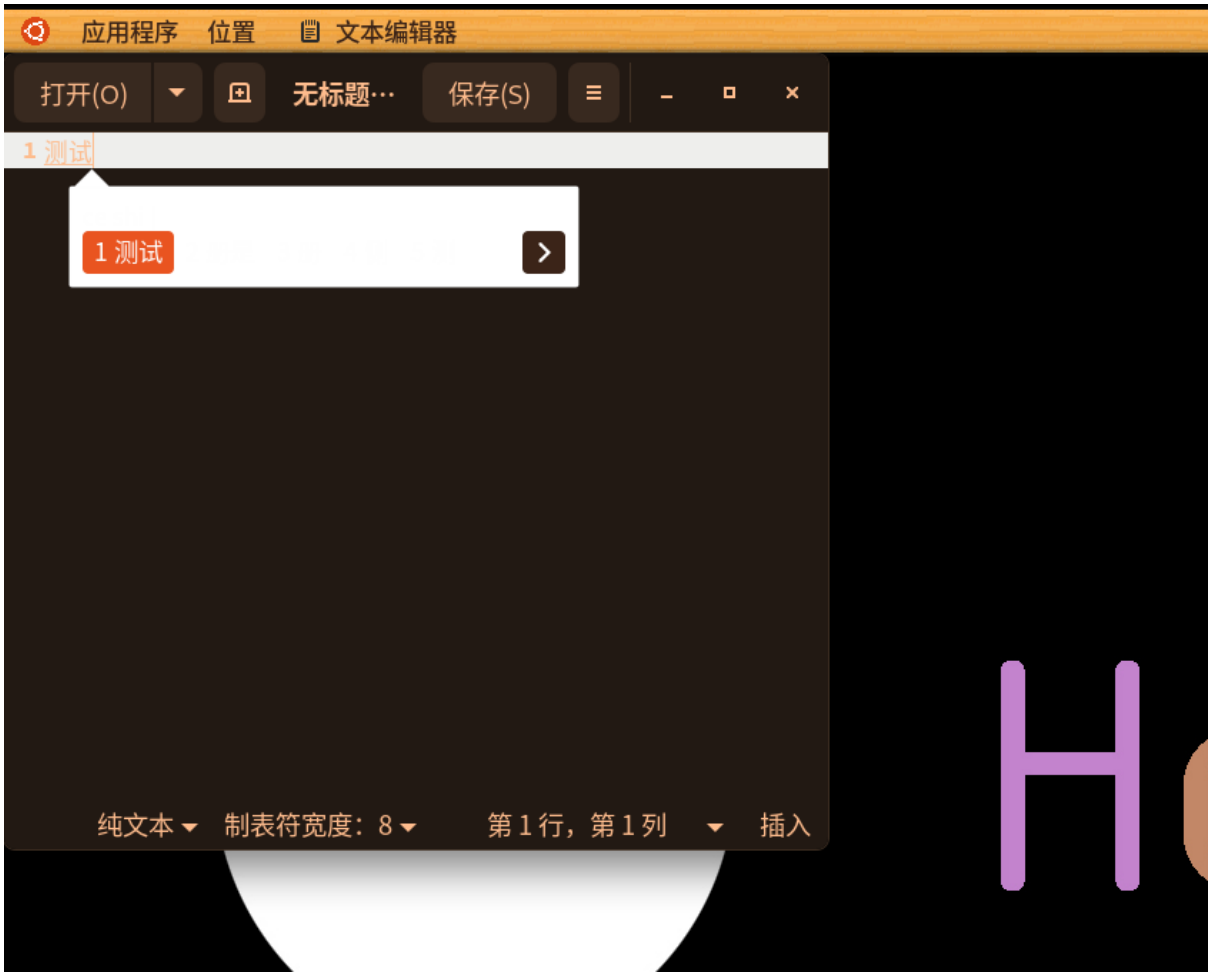
NOTE:

1. Recommend to use X11. If you change themes under Wayland, all your current work may be lost.
2. Tested on Fedora and Ubuntu, GNOME-shell 3.38(v3, v5), 40.0(v4, v6).
3. For users who don't use GNOME but other desktop environments like KDE, the easiest way to change the IBus theme is to use a different GTK theme for IBus.

Seen as the picture below for an example. Users currently choose vim-light-doder for gnome-shell in User Themes, which is a light mode theme, while IBus uses vim-dark-doder theme chosen by users in Customize IBus, which is a dark mode theme. The pre-edit window of ibus-pinyin now uses the dark theme instead of light theme.

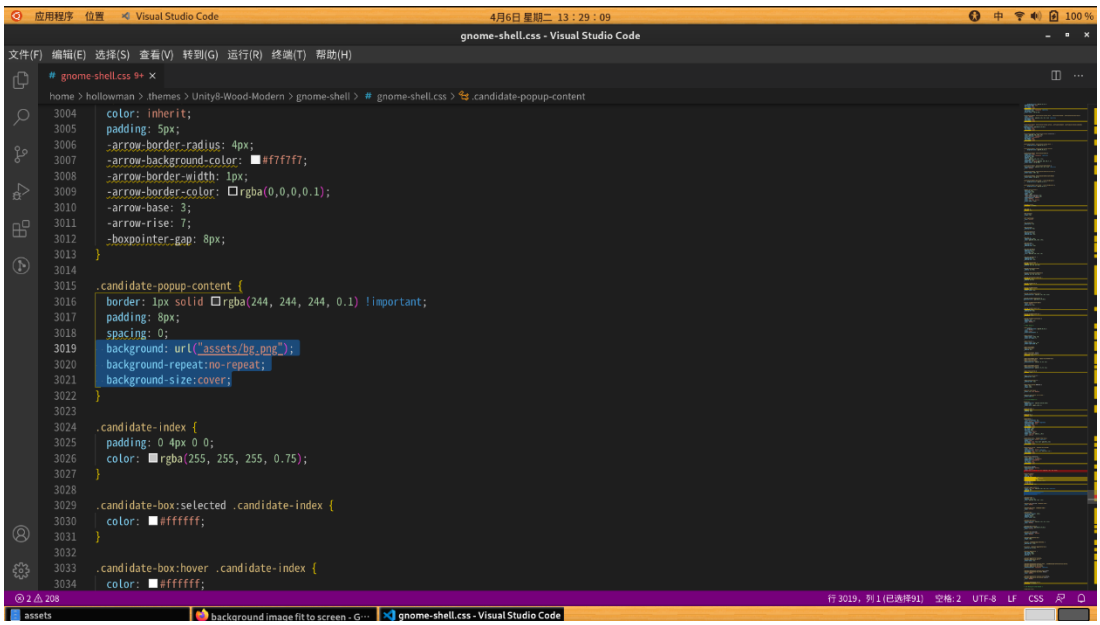


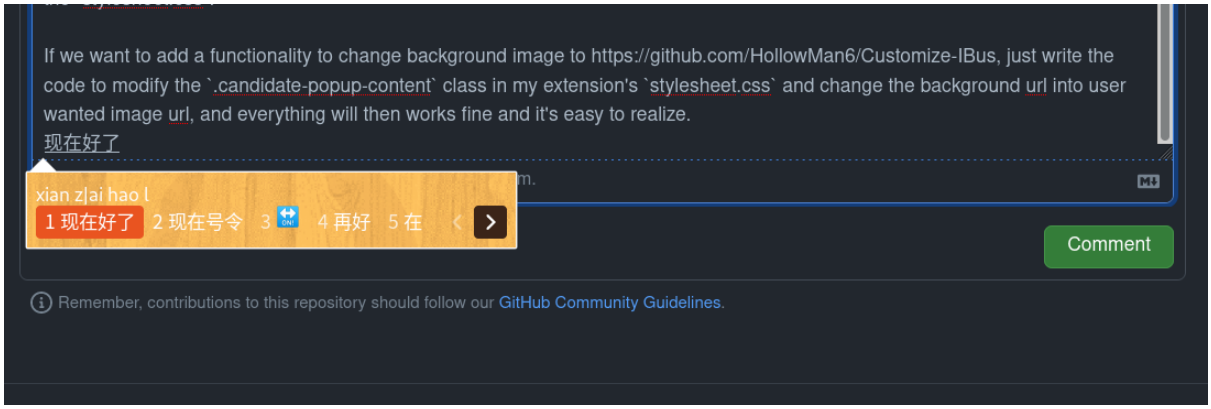
Moreover, when I tried some gnome-shell themes with customized background images, I was using https://www.gnome-look.org/p/1291819/, And it was problematic when I was trying on Fedora 33, GNOME 3.38. Now this gnome-shell theme seems to still be problematic with displaying IBus candidate window correctly and with no background image showing on Fedora rawhide, GNOME 40 changed using the official user theme extension.



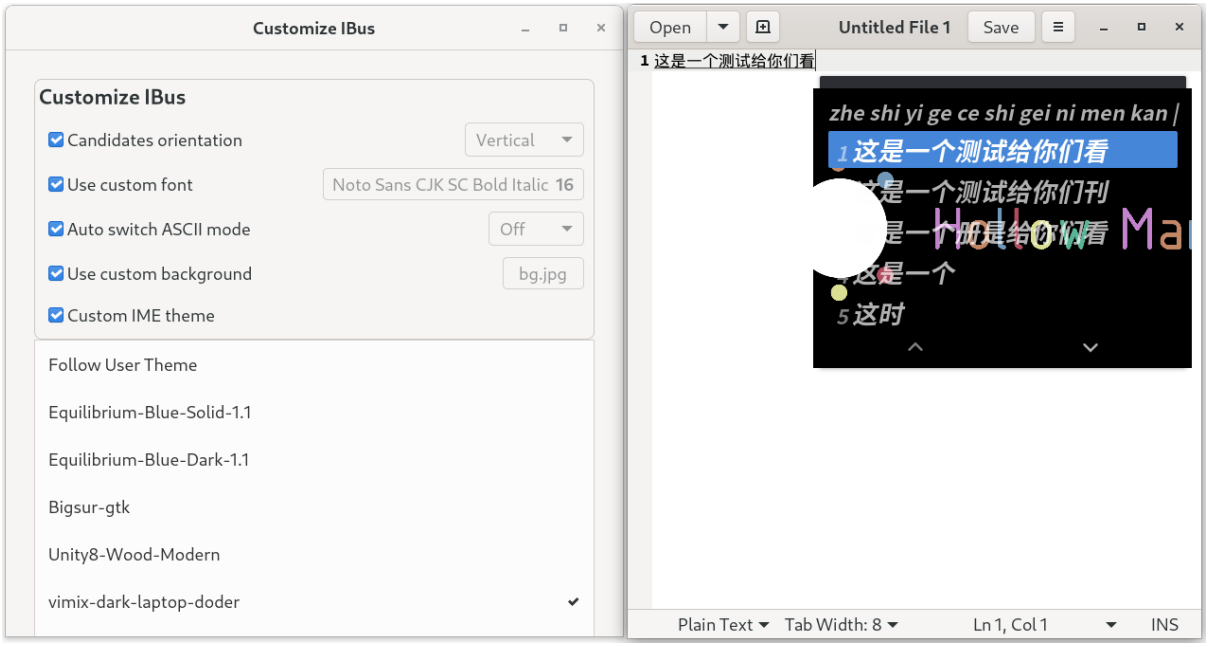
Then I'm able to fix this theme by adding those style code to the gnome-shell.css

```
background: url("assets/bg.png");
background-repeat:no-repeat;
background-size:cover;
```





As a result, I added a custom background image functionality, It's realized by combining ibus-font-setting and background-logo, referring to GNOME-Shell's source code, I found the corresponding widget for class candidate-popup-content, and modified the style, changed the background URL into user wanted image URL to realize modifying IBus background picture without restarting GNOME-Shell.



During the project period, apart from fixing the bugs in the existing demo, I plan to add more features to this GNOME extension so that it can be more useful.

I'll also refactor my code, make theme conversion separate from current extension using Python, extract the IBus related style classes (.candidate-*) by the CSS parser instead of using regular expressions just as currently since it's too ugly and looks too complicated, it may be bugful if the stylesheet is defined in the way that differs, finally write them to another stylesheet specialized for IBus, users can debug it and modify it freely to give users more choices. Then adapt the current extension to accept the user given stylesheet

I've also created a project for changing the IBus GTK theme in a non-GNOME Shell desktop environment: https://github.com/HollowMan6/IBus-Theme , later during GSoC coding period I can merge the above extracting IBus style from GNOME Shell theme function into that too to make a complete project.

4. Timeline and Goals

My university's exam period will begin on June 28, 2021 and end on July 9, 2021. I may work fewer times during that period since I have to prepare for the exams. In the rest of time, I can work full time (unlimited time, at least 20 hours per week) because I have no other commitments except for the exam period.

I know a mentor isn't a replacement for Google or stack overflow. I will only discuss the difficulties that I can't overcome by myself with mentors and send a "summary of the week" every week by email to mentors to inform my progress of the work so far and what I have achieved this week.

Here is the timeline:

A. Community Bonding period (May 18, 2021 - June 8, 2021)

During the community bonding period, I aim to go through the IBus codebase thoroughly. Currently, I already have some knowledge about the codebase. However, going through the entire codebase will help me work faster during my coding period and make it possible to contribute to other features of the project in the future.

I would also like to interact with the mentors as well as get to know the community as a whole and help out whenever I can.

I expect to try my best right after the start of the community bonding period instead of waiting for the coding period to start.

B. First Part of Coding Time (June 8, 2021 – July 13, 2021)

- IBus customized pre-edit themes.
- Help out in the community.
- Other tasks instructed by the mentors.

C. First Evaluation Phase (July 13 - 17, 2021)

Deliverables

- Finished IBus customized pre-edit themes.

D. Final Part of Coding Time (July 18, 2021 - August 17, 2021)

- Time for delays in IBus customized pre-edit themes.
- Help out in the community.
- Other tasks instructed by the mentors.

E. Final Evaluation Phase (August 17 - 24, 2021)

Deliverables

- Finished IBus customized pre-edit themes.
- Finished other tasks instructed by the mentors.

F. Future (September 1, 2021 -)

I will continue contributing and do my best to maintain the IBus project after I successfully complete the GSoC project and if possible, I am happy to be a mentor of IBus project in the future GSoC.