Art2Dec®

Mel (<Chalk> Мел) Editor – MiniEditor for Linux.

Igor Lukyanov

February, 2025

Introduction.

Mel (Mini Embedded Light Editor for Linux) is a terminal-based text editor designed for Linux and Unix platforms.

Written in C from scratch, Mel is highly minimalistic, dependency-independent, optimized for environments where resources are constrained.

Syntax highlighting for several programming languages and support for various platforms is already provided.

Mel aims to provide a lightweight yet powerful solution for developers working across diverse environments.

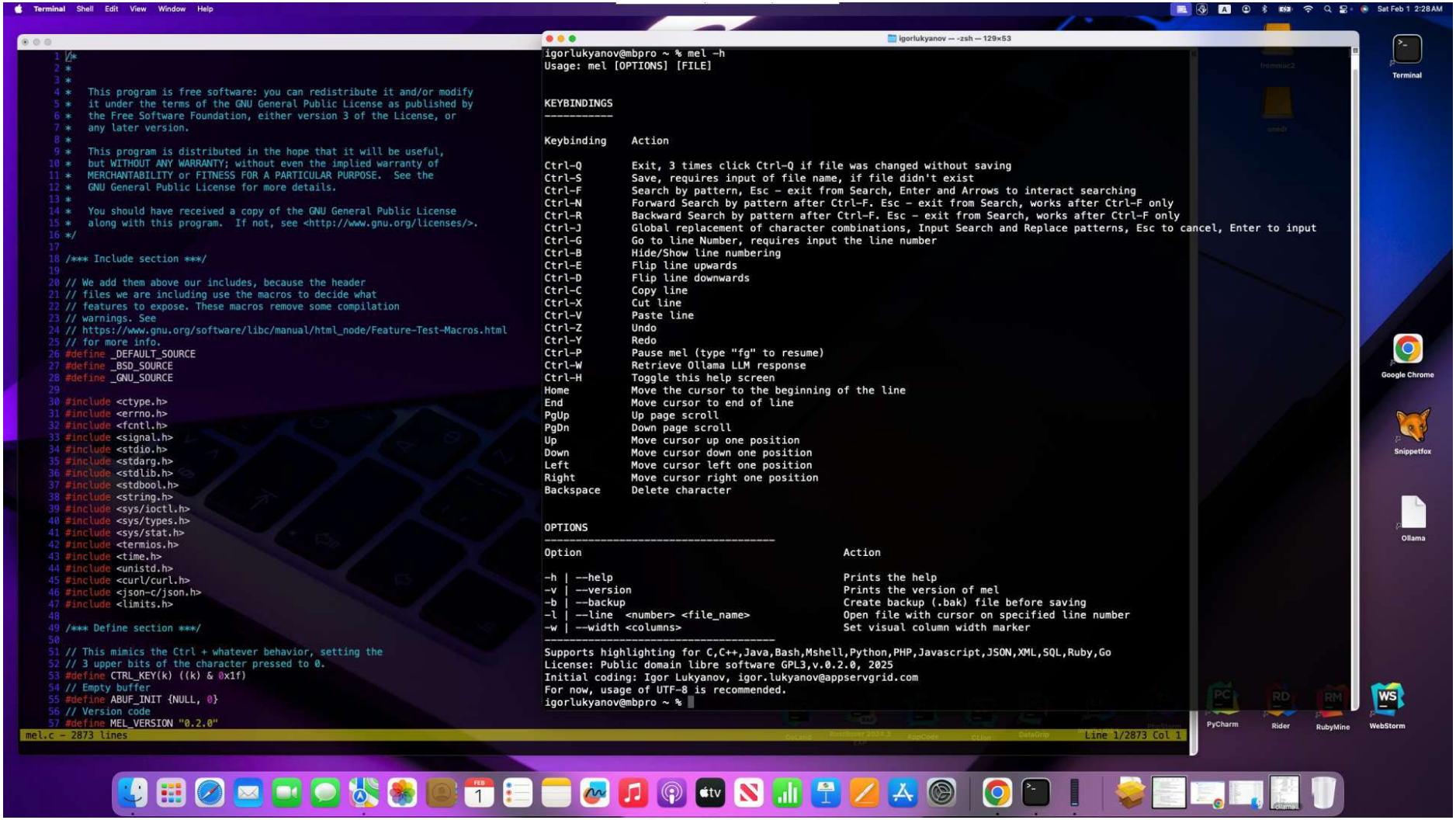
Main Goals

- To create a lightweight, minimalistic multiplatform text editor with no dependencies.
- To support resource-intensive environments such as Kubernetes pods, Docker containers, and IoT devices.
- To provide syntax highlighting for multiple programming languages.
- To ensure compatibility with ARM64 architectures and other CPU types (Intel, AMD, etc.) and different Oses.

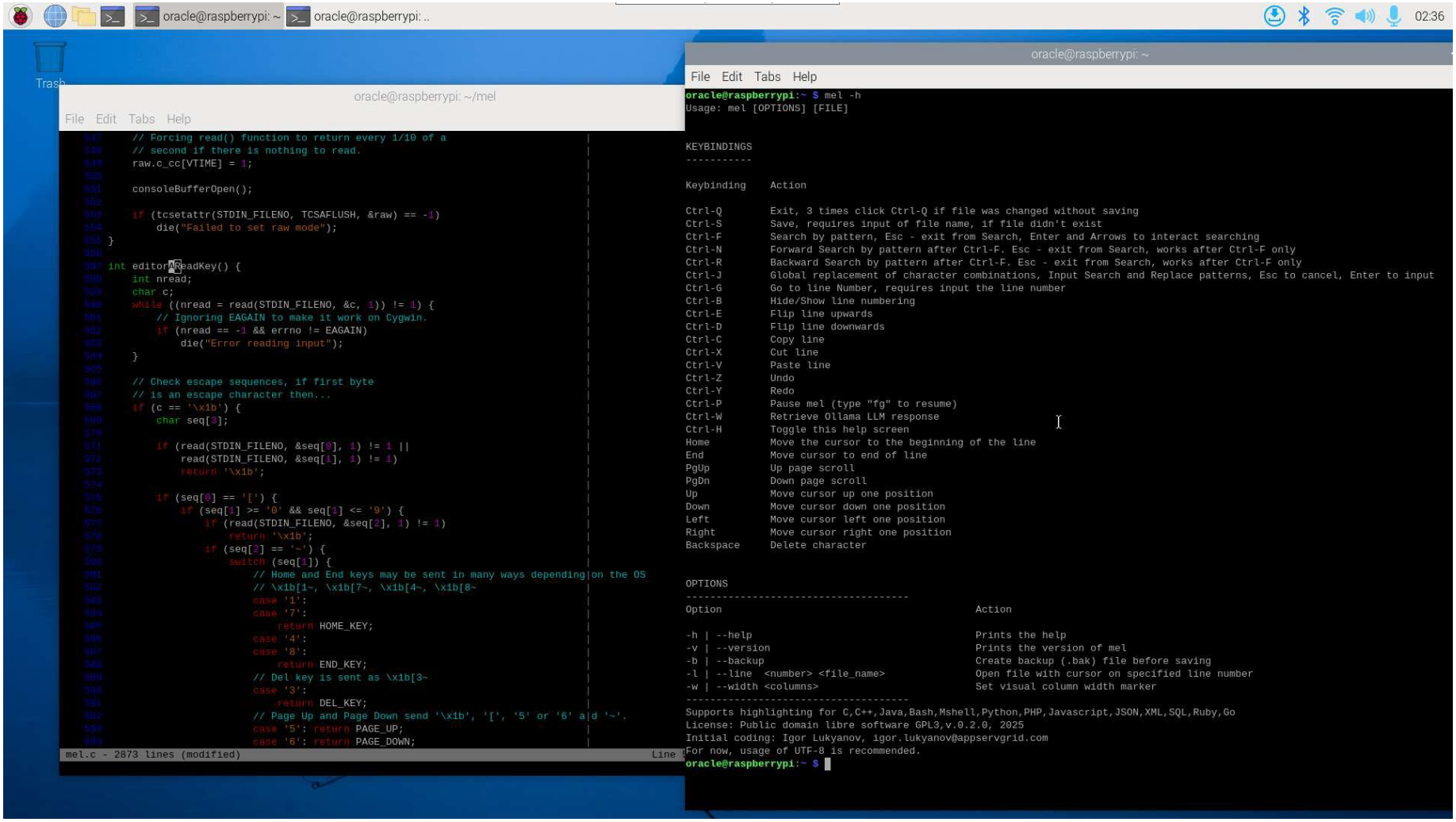
Key features.

- Multi-platform Support:
- Works seamlessly on Ubuntu Linux, Raspberry Pi OS, and macOS and it was tested on them.
- Supports various architectures, including ARM64 and x86_64, Apple M1, etc.
- Keybindings.
- Efficient keyboard shortcuts for navigation, editing, and functionality (refer to CLI help).
- Minimalistic Design.
- Dependency-free (does not rely on libraries like curses).

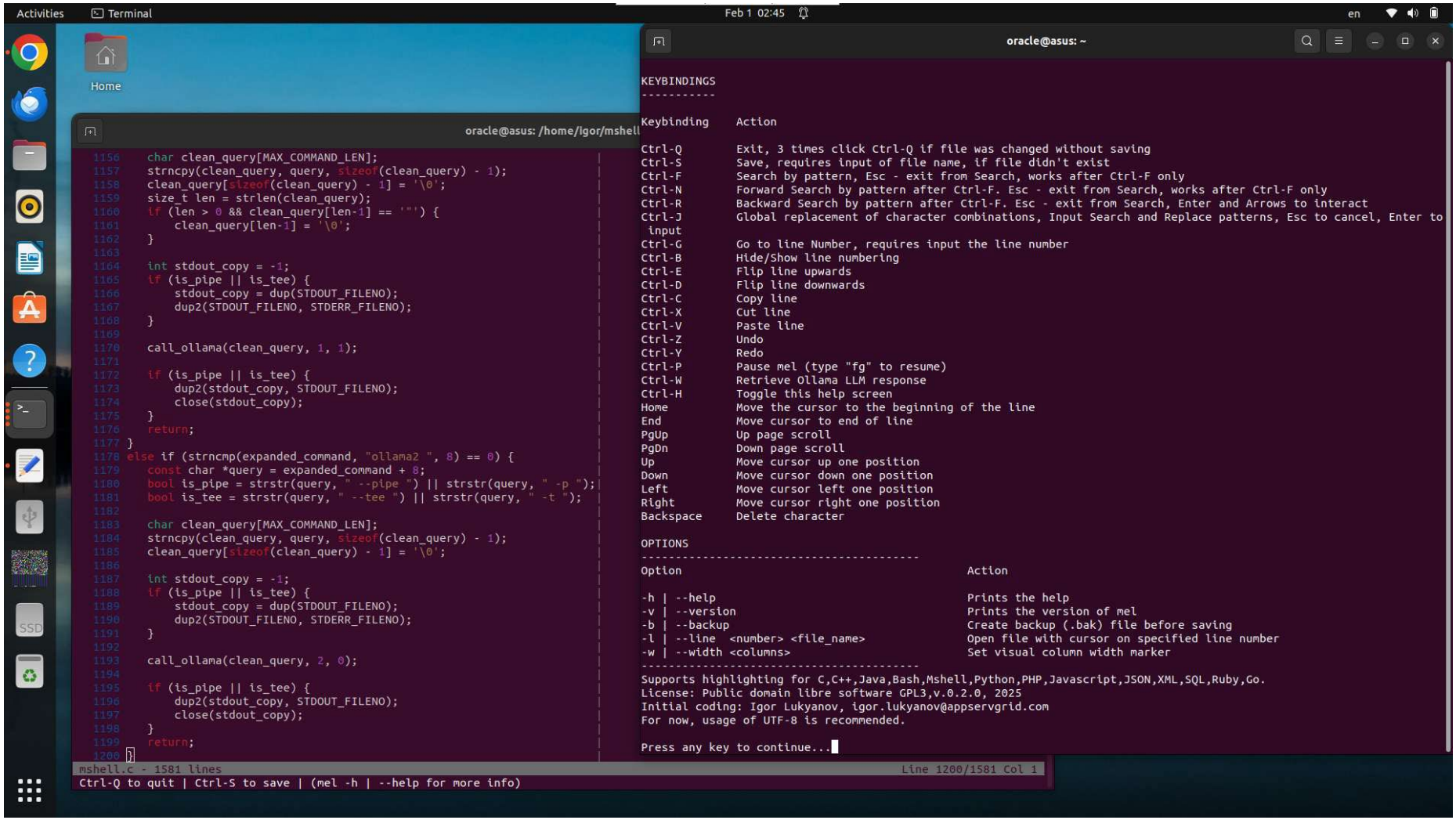
MacOS platform.



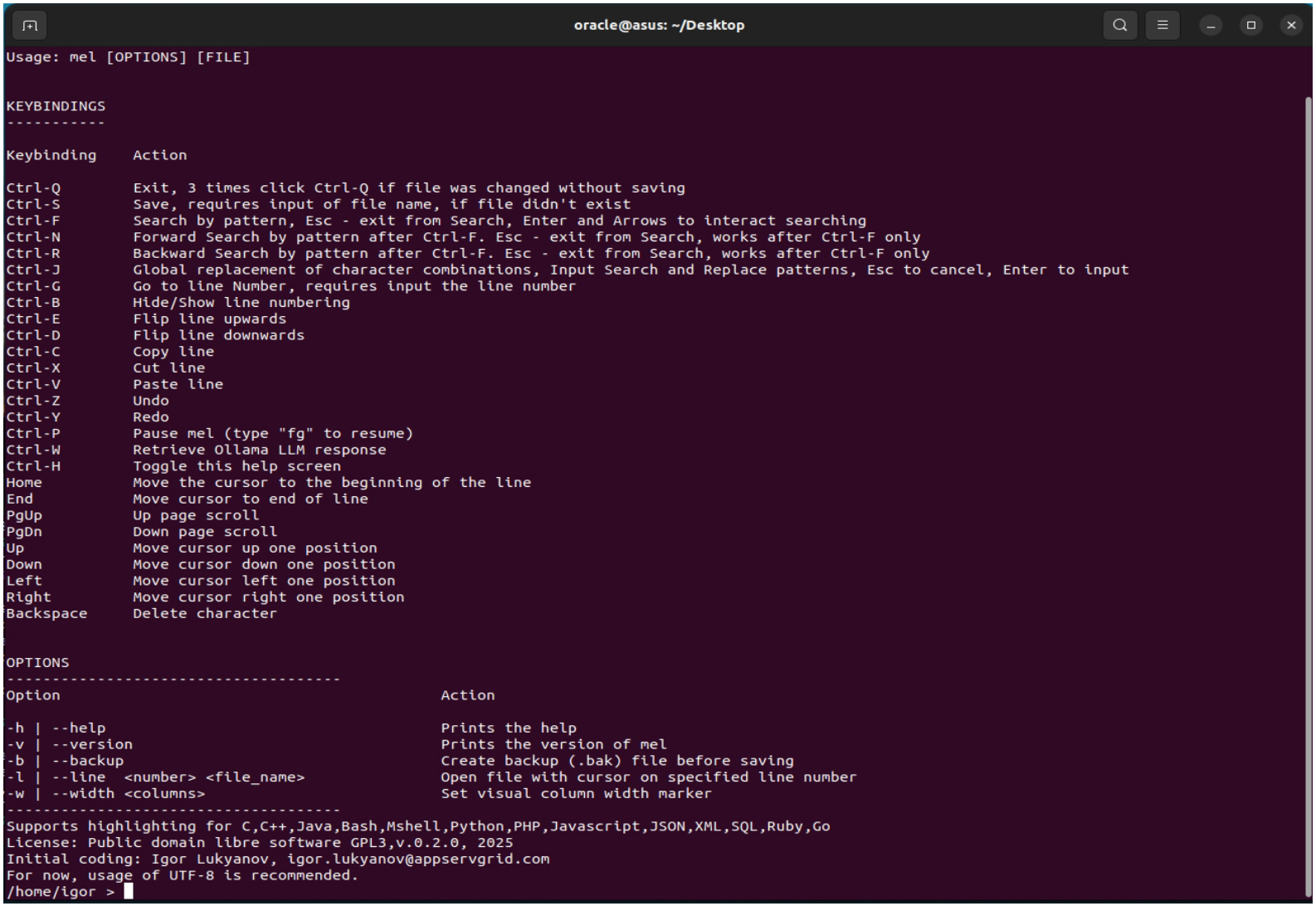
Raspberry PI OS.



Ubuntu Linux 2x.0x OS and other Linux like Debian, Linux Mint, CentOS, Fedora, Oracle Linux, RedHat EL, etc.



CLI help screen with Keybindings and start Options



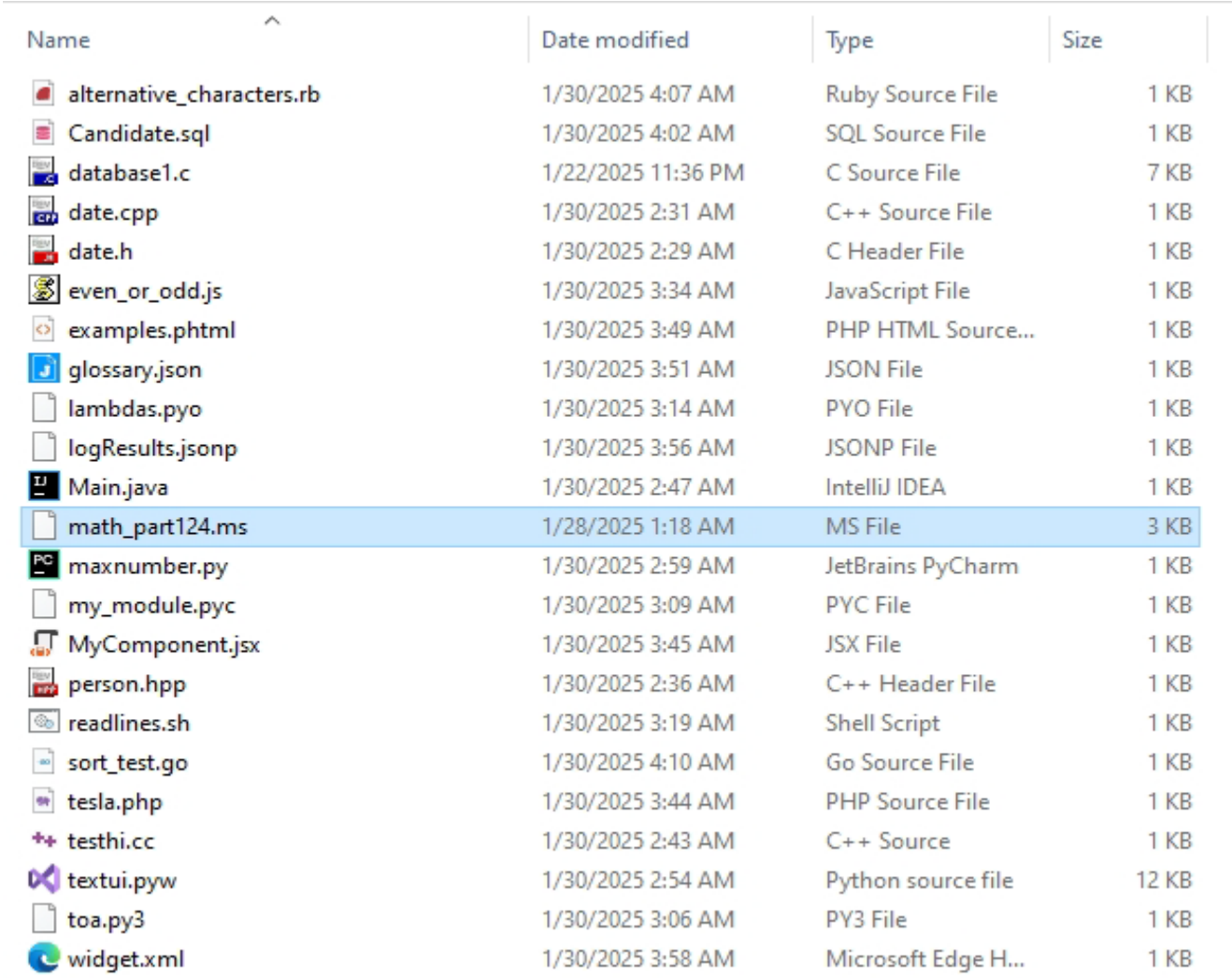
Modern current supported version.

```
oracle@asus:~/Desktop$ mel --version
mel - version 0.2.0
oracle@asus:~/Desktop$
```

Highlighting Supported Languages.

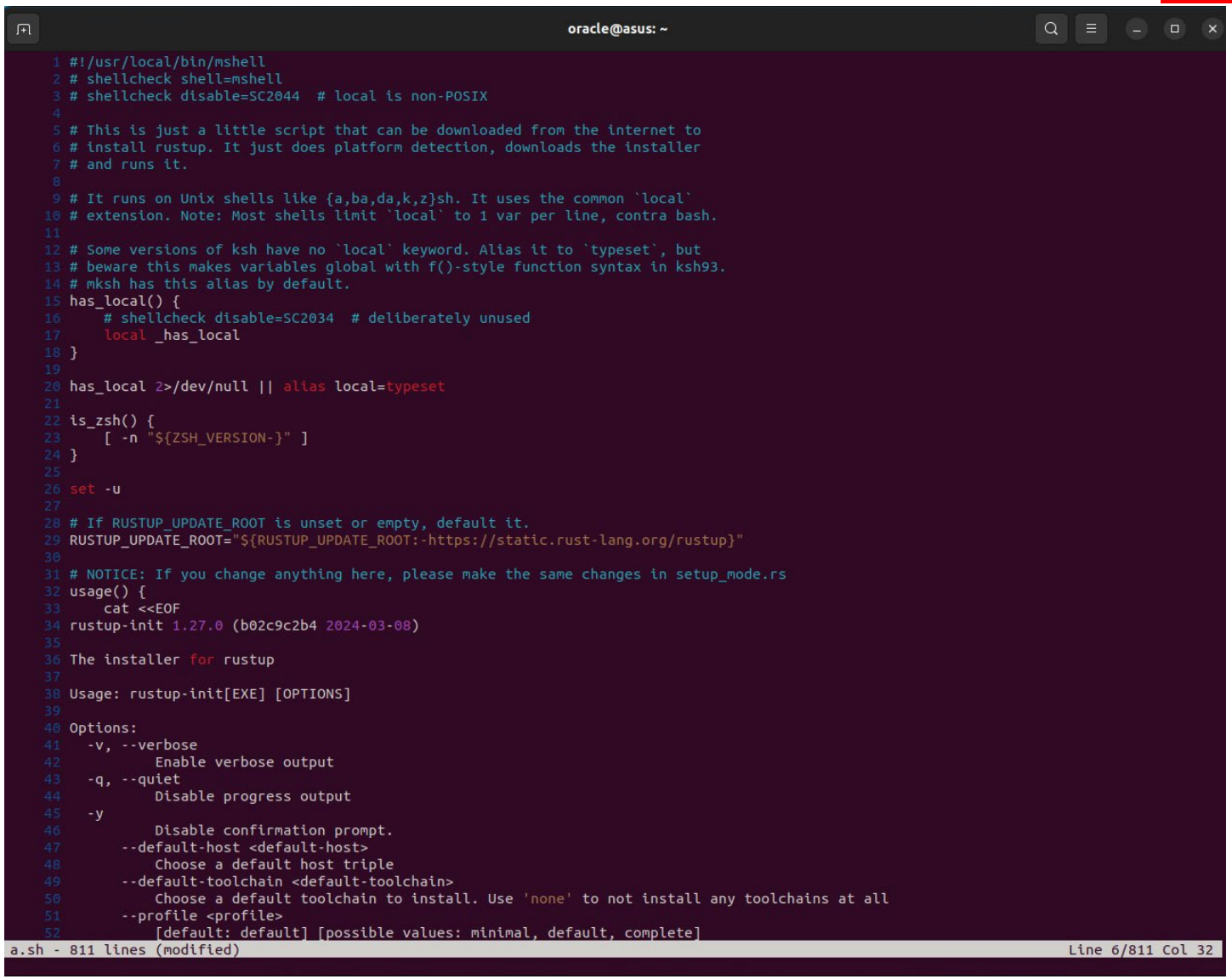
- Mel supports syntax highlighting for a wide range of programming languages:
- C, C++, Java, Bash, Mshell, Python, PHP
- JavaScript, JSON, XML, SQL, Ruby, Go
- Etc. It is also possible to add data syntax format for highlighting by your request.

Main data formats for syntax highligting.

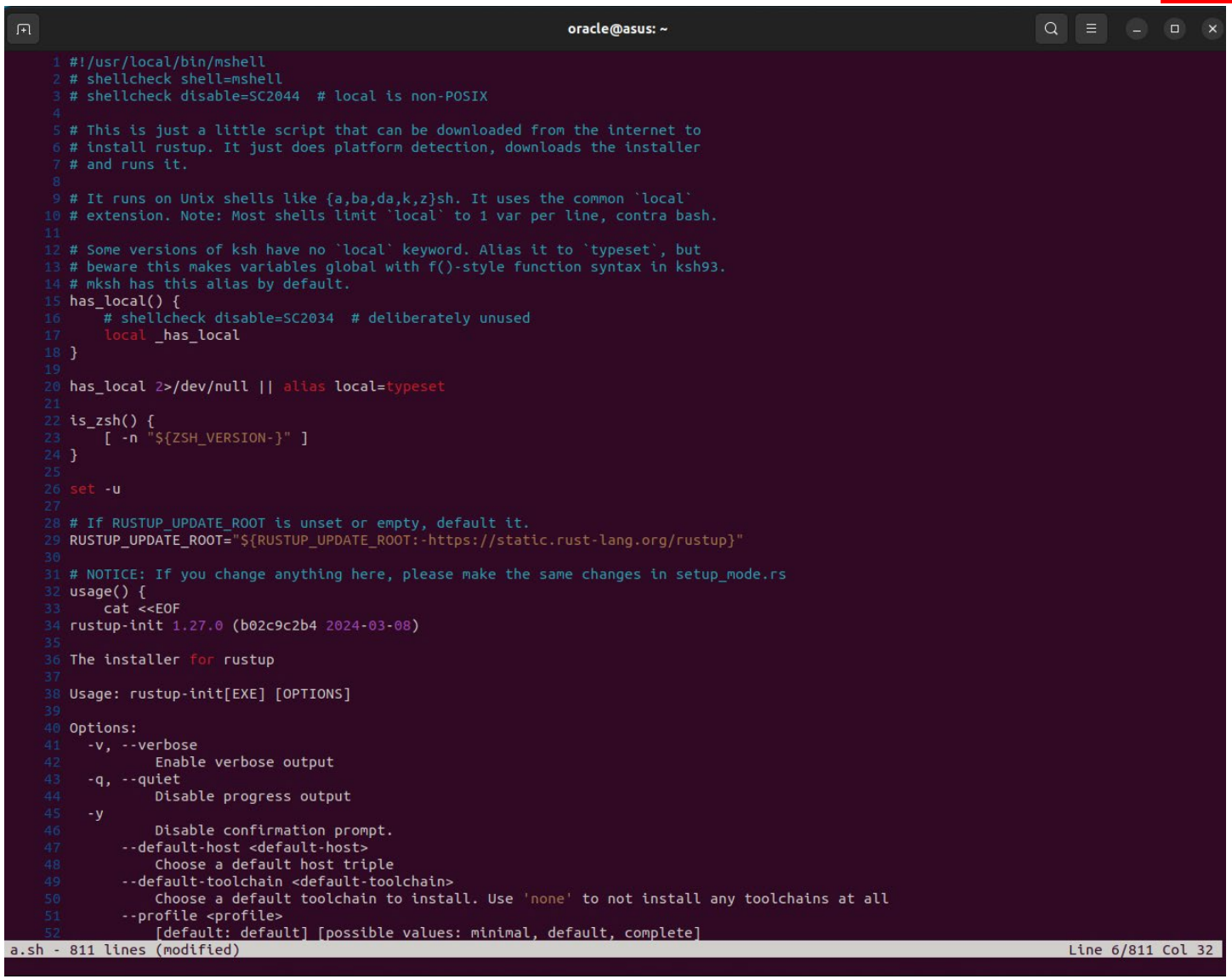


Name	Date modified	Type	Size
alternative_characters.rb	1/30/2025 4:07 AM	Ruby Source File	1 KB
Candidate.sql	1/30/2025 4:02 AM	SQL Source File	1 KB
database1.c	1/22/2025 11:36 PM	C Source File	7 KB
date.cpp	1/30/2025 2:31 AM	C++ Source File	1 KB
date.h	1/30/2025 2:29 AM	C Header File	1 KB
even_or_odd.js	1/30/2025 3:34 AM	JavaScript File	1 KB
examples.phtml	1/30/2025 3:49 AM	PHP HTML Source...	1 KB
glossary.json	1/30/2025 3:51 AM	JSON File	1 KB
lambdas.pyo	1/30/2025 3:14 AM	PYO File	1 KB
logResults.jsonp	1/30/2025 3:56 AM	JSONP File	1 KB
Main.java	1/30/2025 2:47 AM	IntelliJ IDEA	1 KB
math_part124.ms	1/28/2025 1:18 AM	MS File	3 KB
maxnumber.py	1/30/2025 2:59 AM	JetBrains PyCharm	1 KB
my_module.pyc	1/30/2025 3:09 AM	PYC File	1 KB
MyComponent.jsx	1/30/2025 3:45 AM	JSX File	1 KB
person.hpp	1/30/2025 2:36 AM	C++ Header File	1 KB
readlines.sh	1/30/2025 3:19 AM	Shell Script	1 KB
sort_test.go	1/30/2025 4:10 AM	Go Source File	1 KB
tesla.php	1/30/2025 3:44 AM	PHP Source File	1 KB
testhi.cc	1/30/2025 2:43 AM	C++ Source	1 KB
textui.pyw	1/30/2025 2:54 AM	Python source file	12 KB
toa.py3	1/30/2025 3:06 AM	PY3 File	1 KB
widget.xml	1/30/2025 3:58 AM	Microsoft Edge H...	1 KB

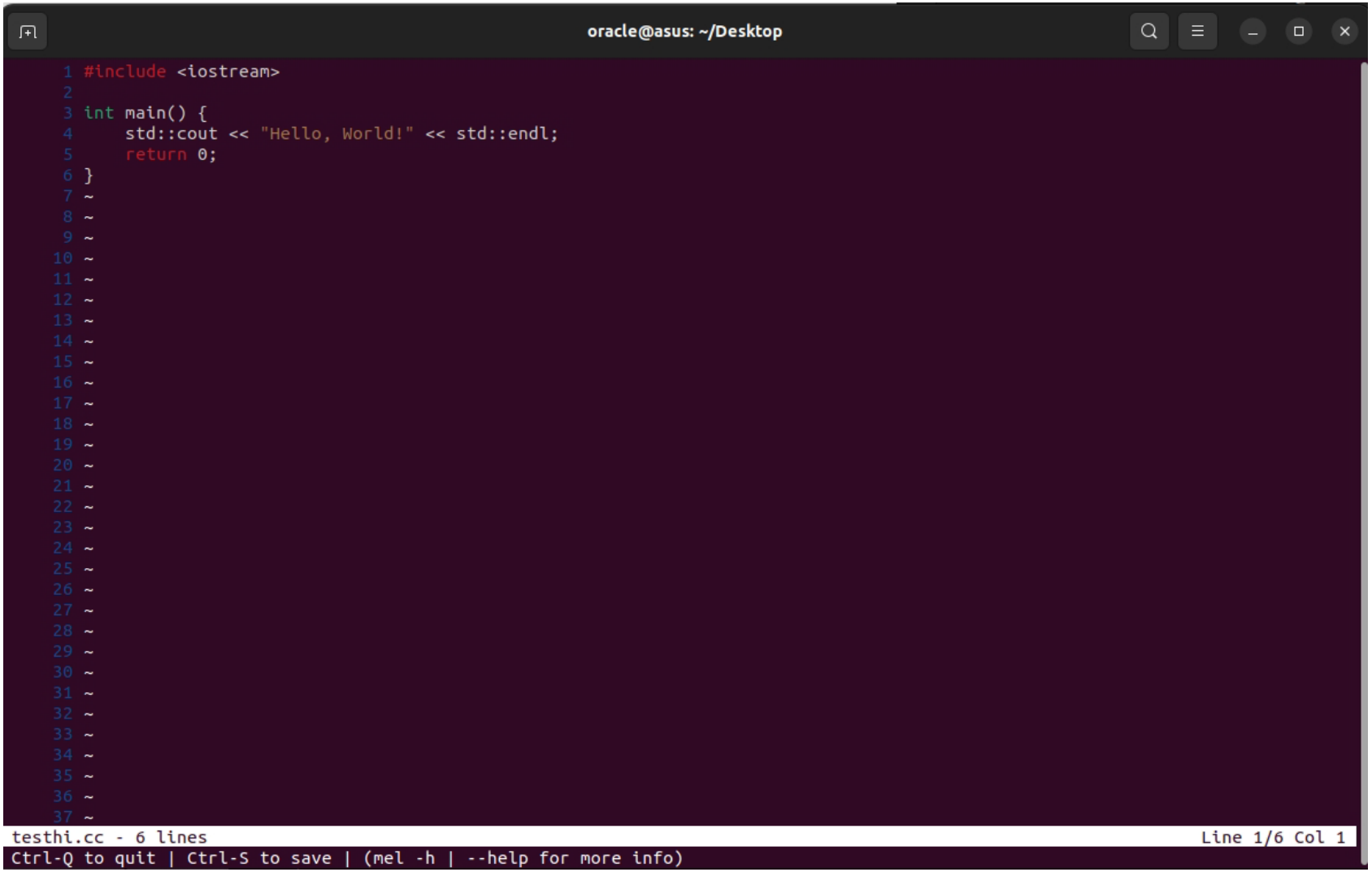
Mshell highlighting support.



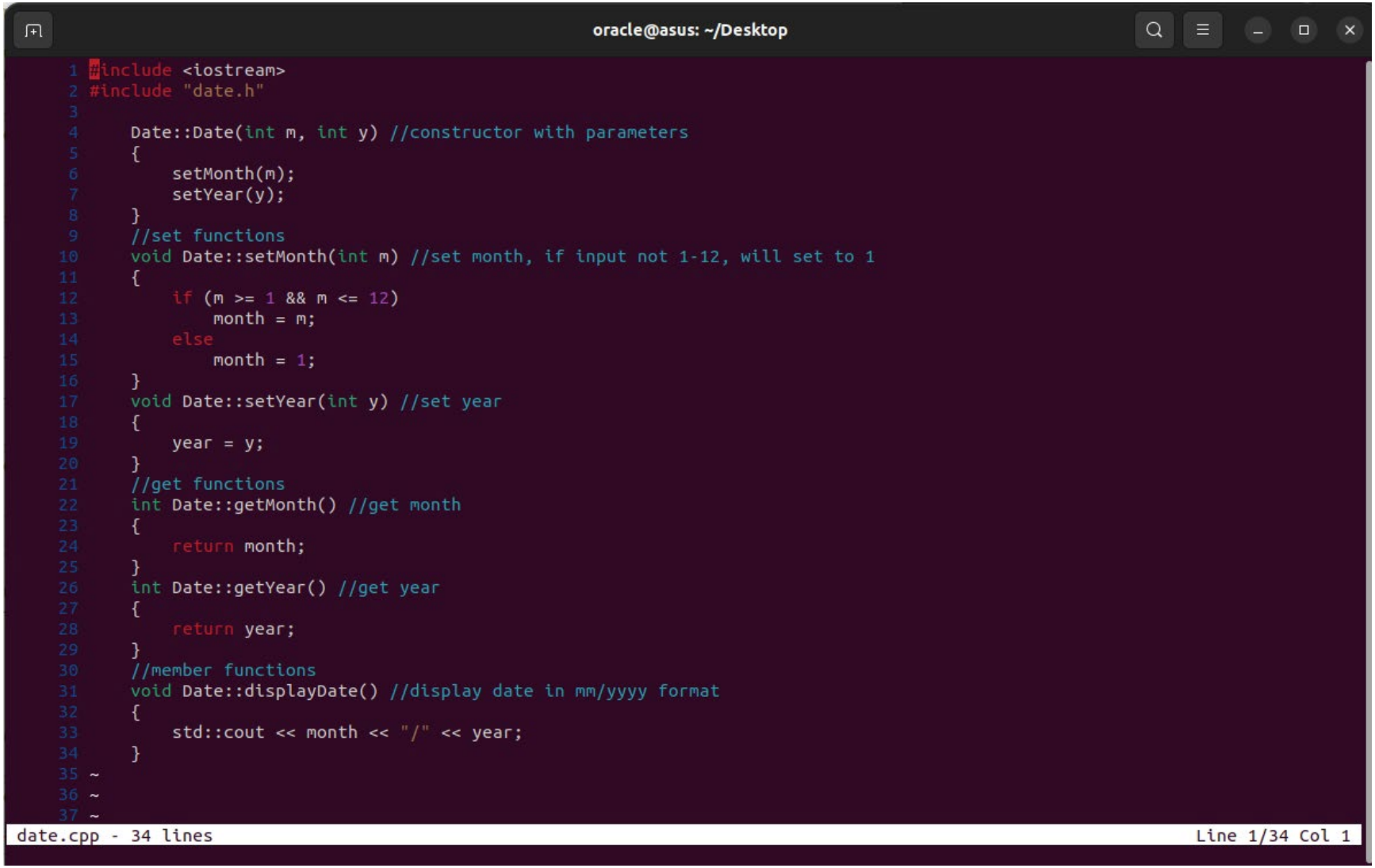
Bash highlighting support.



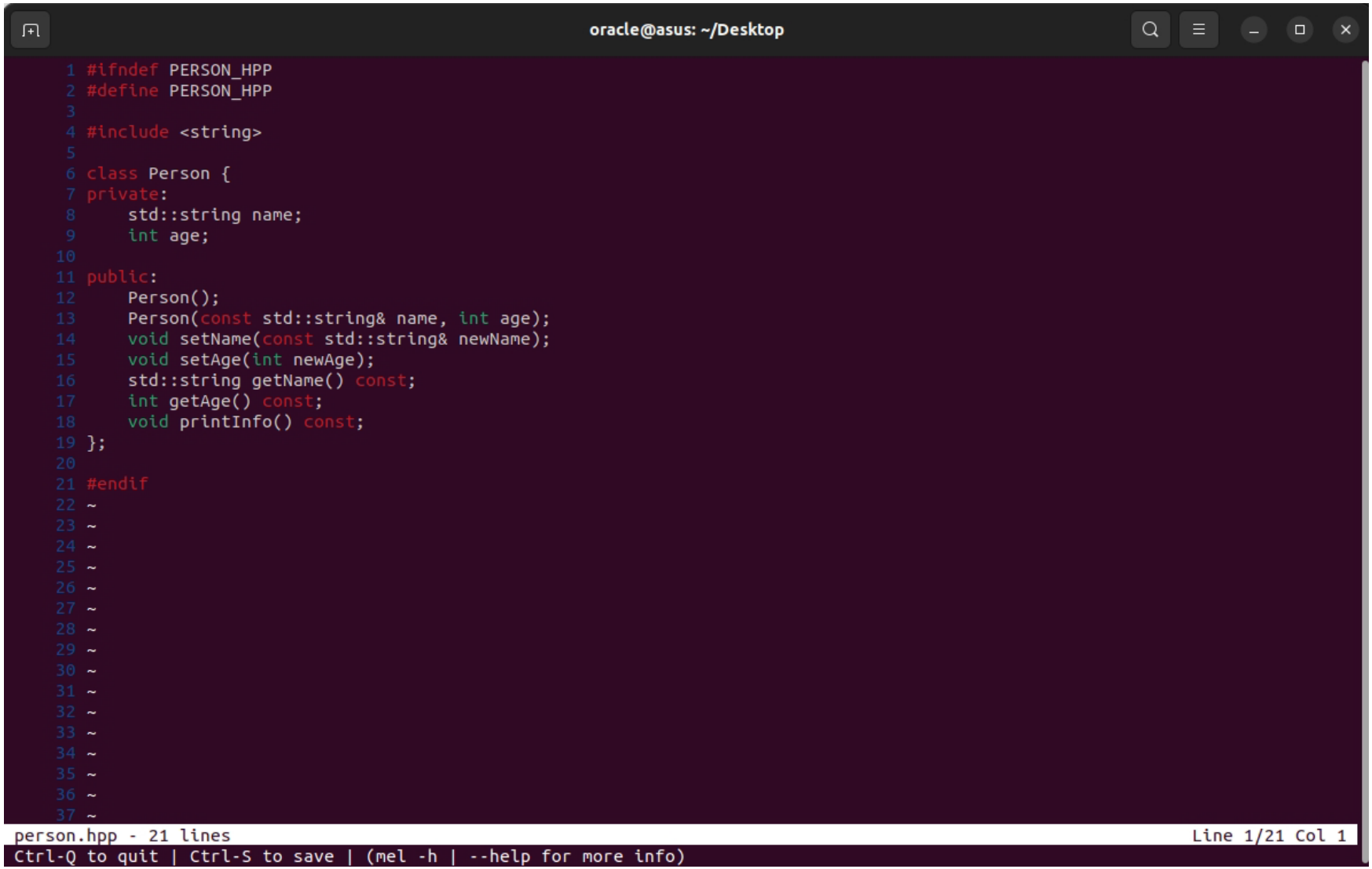
C++_cc highlighting support.



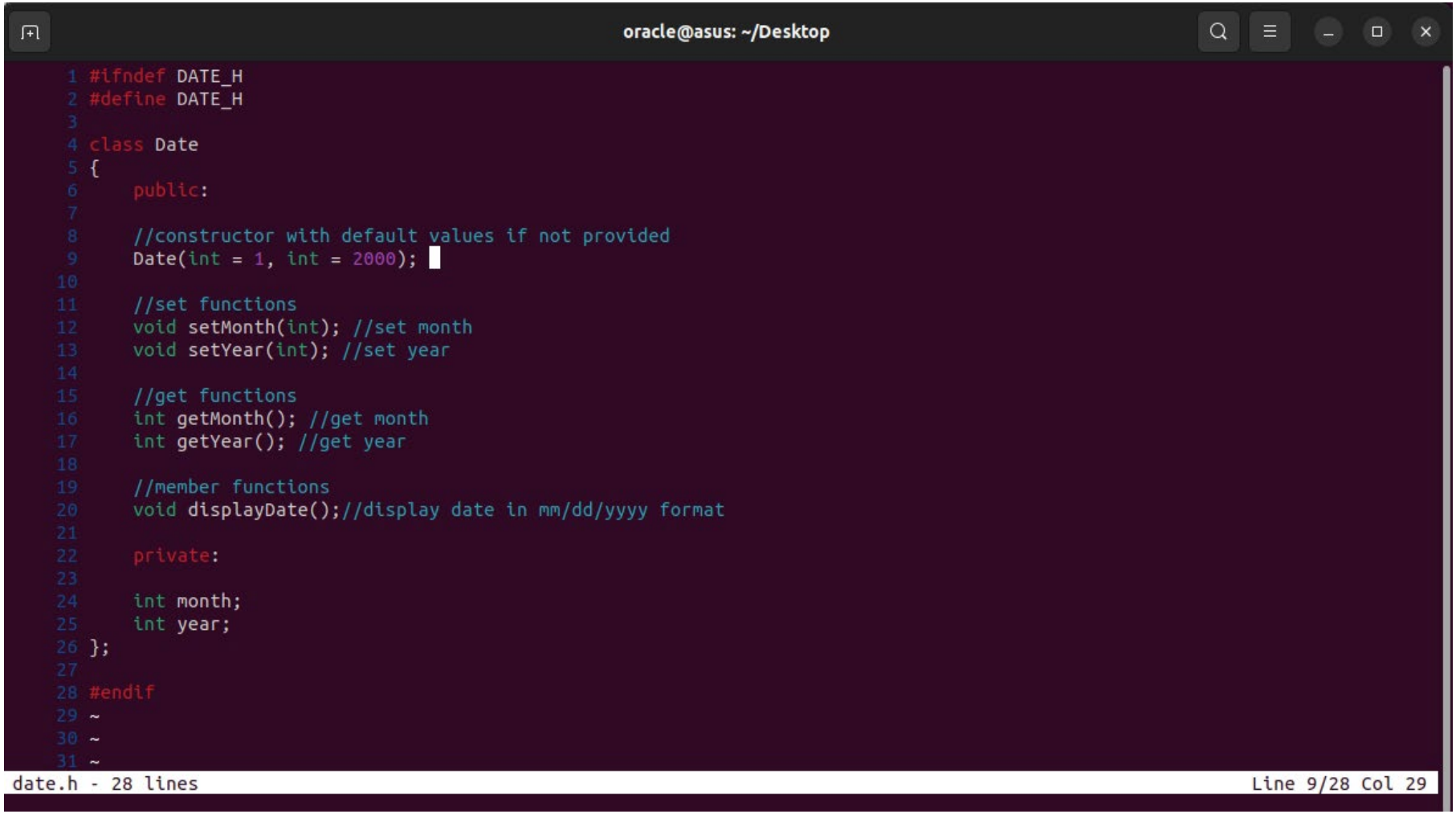
C++_cpp highlighting support.



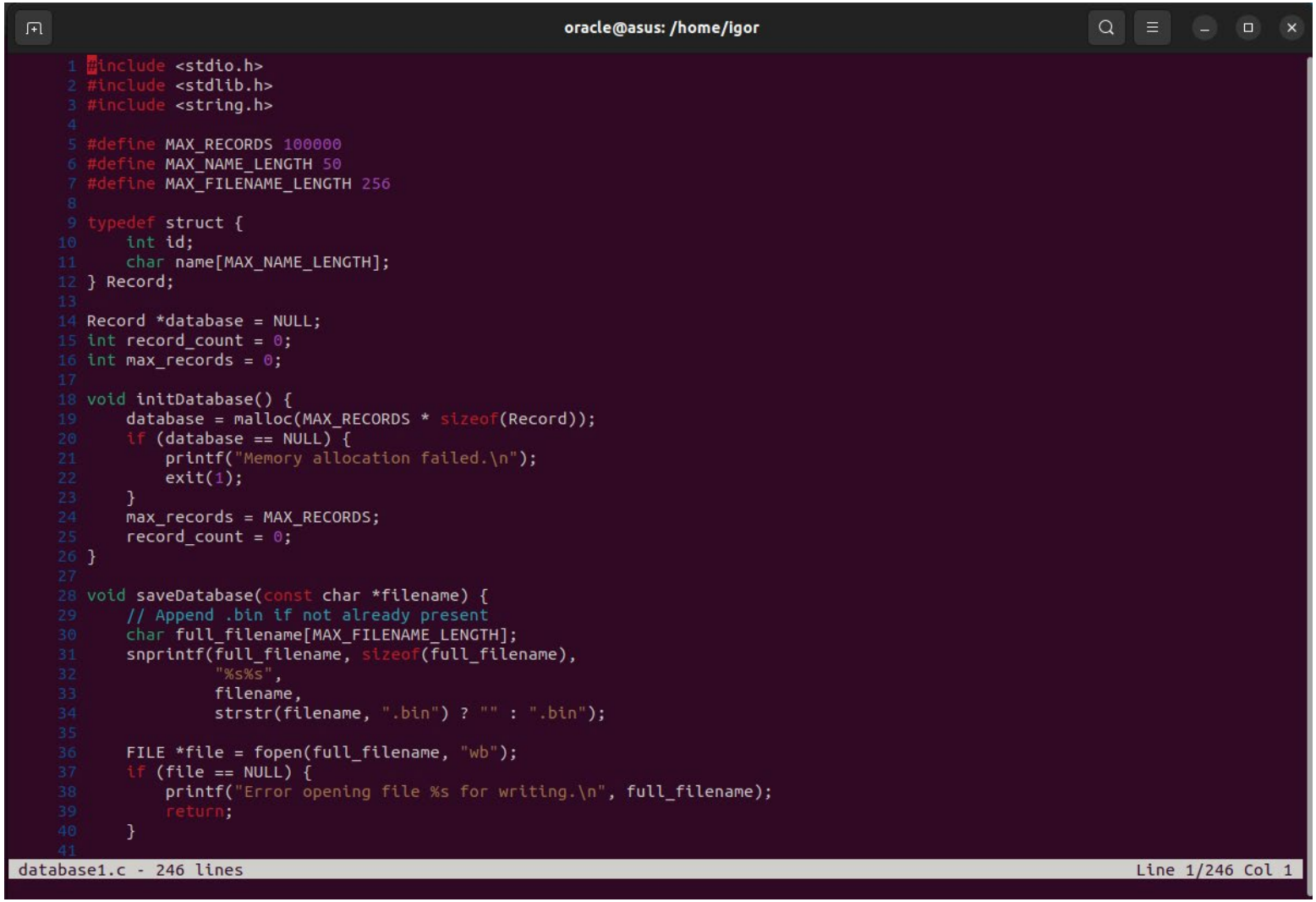
C++_hpp header highlighting support.



C++ header highlighting support.

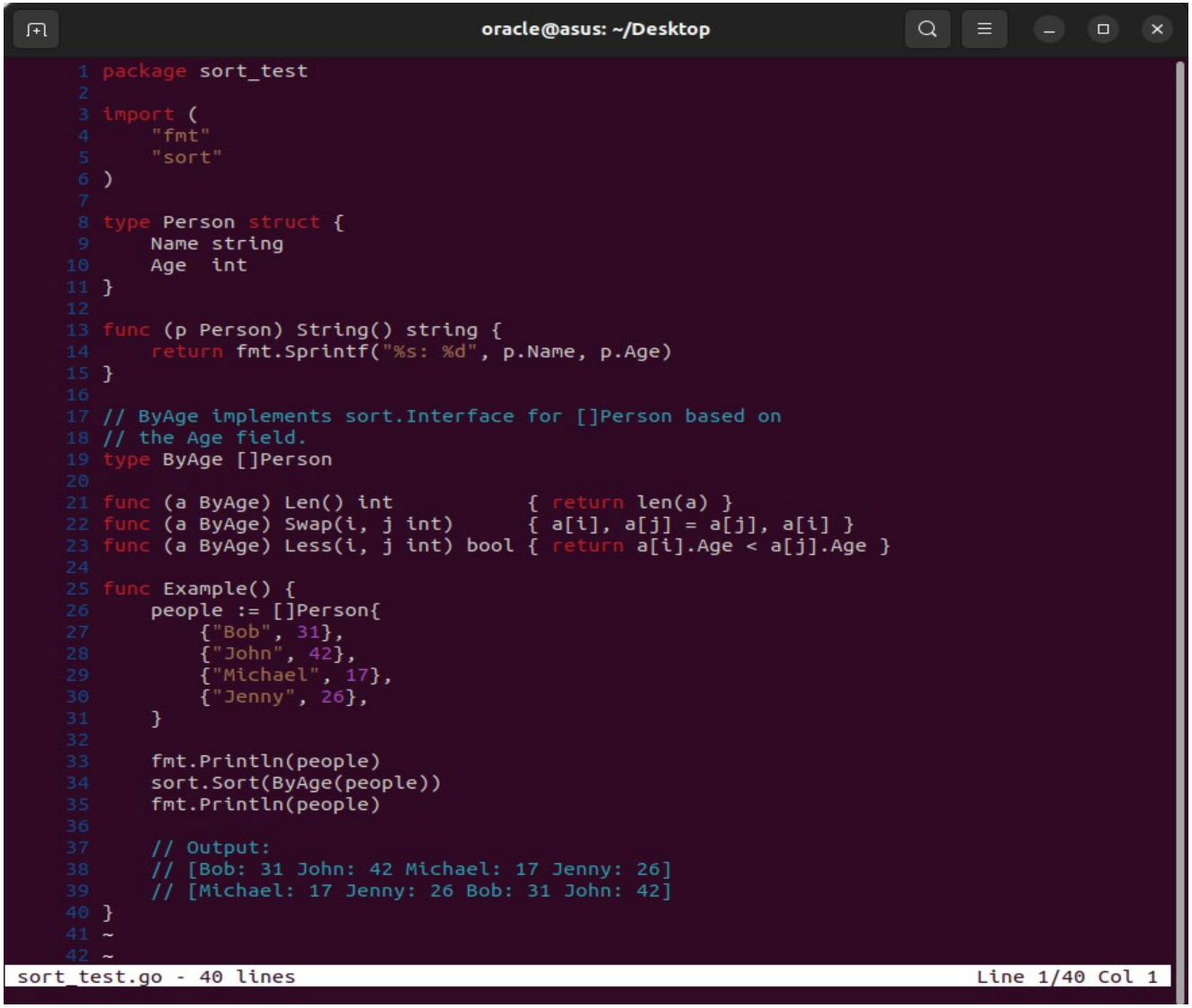


C lang highlighting support (ANSI C, C89 (ANSI X3.159-1989), ISO C, C90 (ISO/IEC 9899:1990), C99, C9X (ISO/IEC 9899:1999), C11, C1X (ISO/IEC 9899:2011), C17, C18 (ISO/IEC 9899:2018), C23, C2X (ISO/IEC 9899:2024)}.

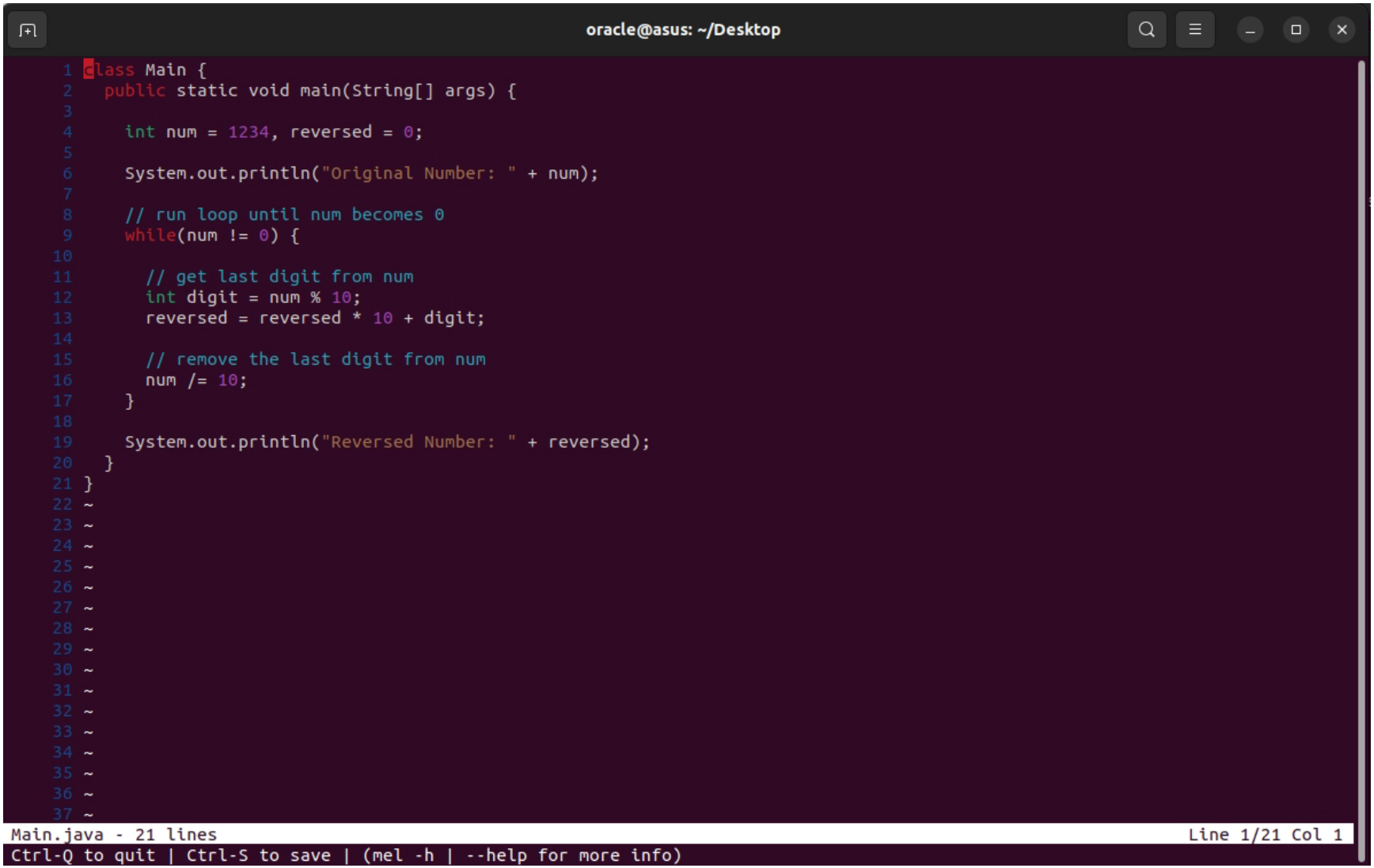



```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define MAX_RECORDS 100000
#define MAX_NAME_LENGTH 50
#define MAX_FILENAME_LENGTH 256

typedef struct {
    int id;
    char name[MAX_NAME_LENGTH];
} Record;

Record *database = NULL;
int record_count = 0;
int max_records = 0;

void initDatabase() {
    database = malloc(MAX_RECORDS * sizeof(Record));
    if (database == NULL) {
        printf("Memory allocation failed.\n");
        exit(1);
    }
    max_records = MAX_RECORDS;
    record_count = 0;
}

void saveDatabase(const char *filename) {
    // Append .bin if not already present
    char full_filename[MAX_FILENAME_LENGTH];
    snprintf(full_filename, sizeof(full_filename),
             "%s%s",
             filename,
             strstr(filename, ".bin") ? "" : ".bin");

    FILE *file = fopen(full_filename, "wb");
    if (file == NULL) {
        printf("Error opening file %s for writing.\n", full_filename);
        return;
    }

```

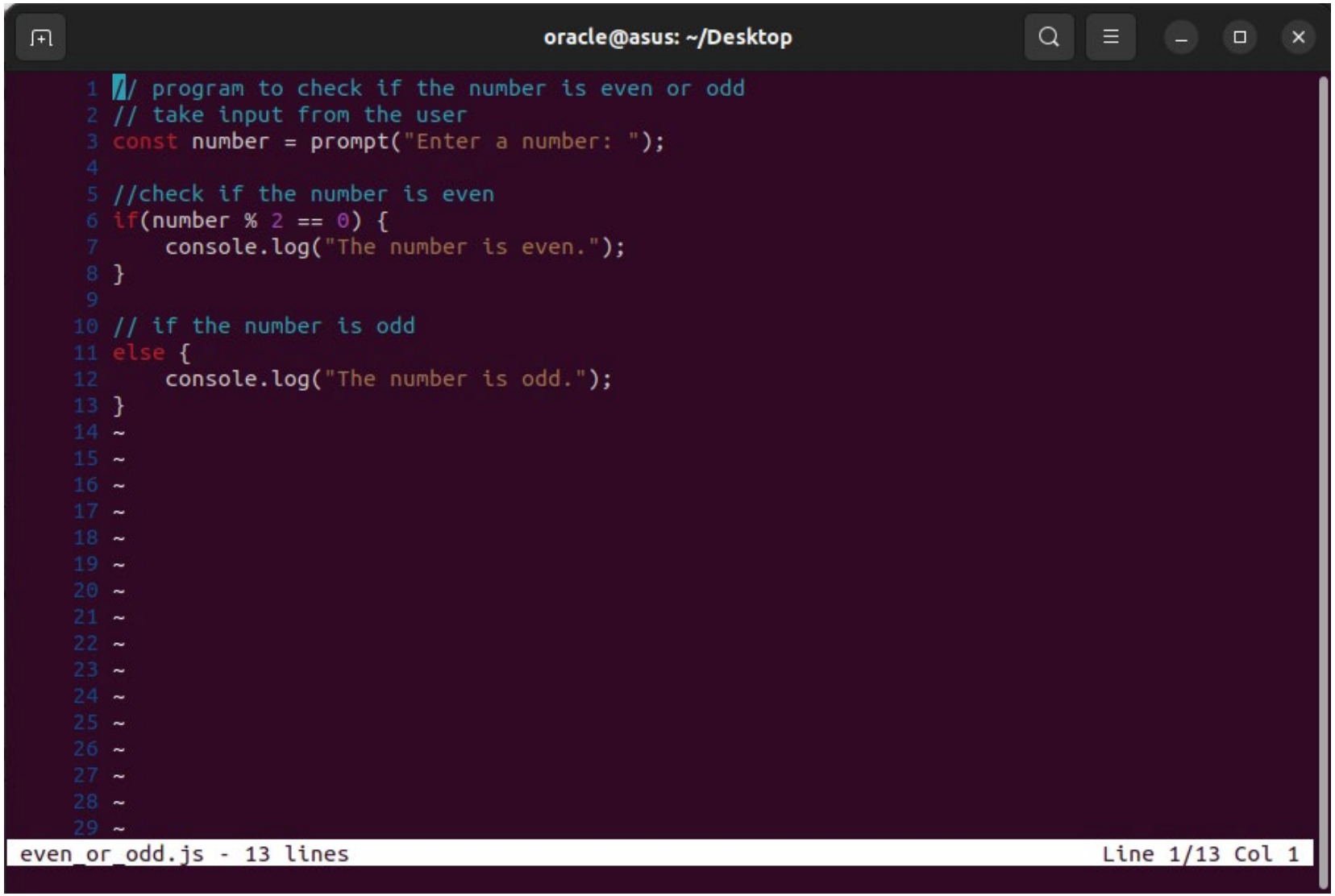
Go language highlighting support.



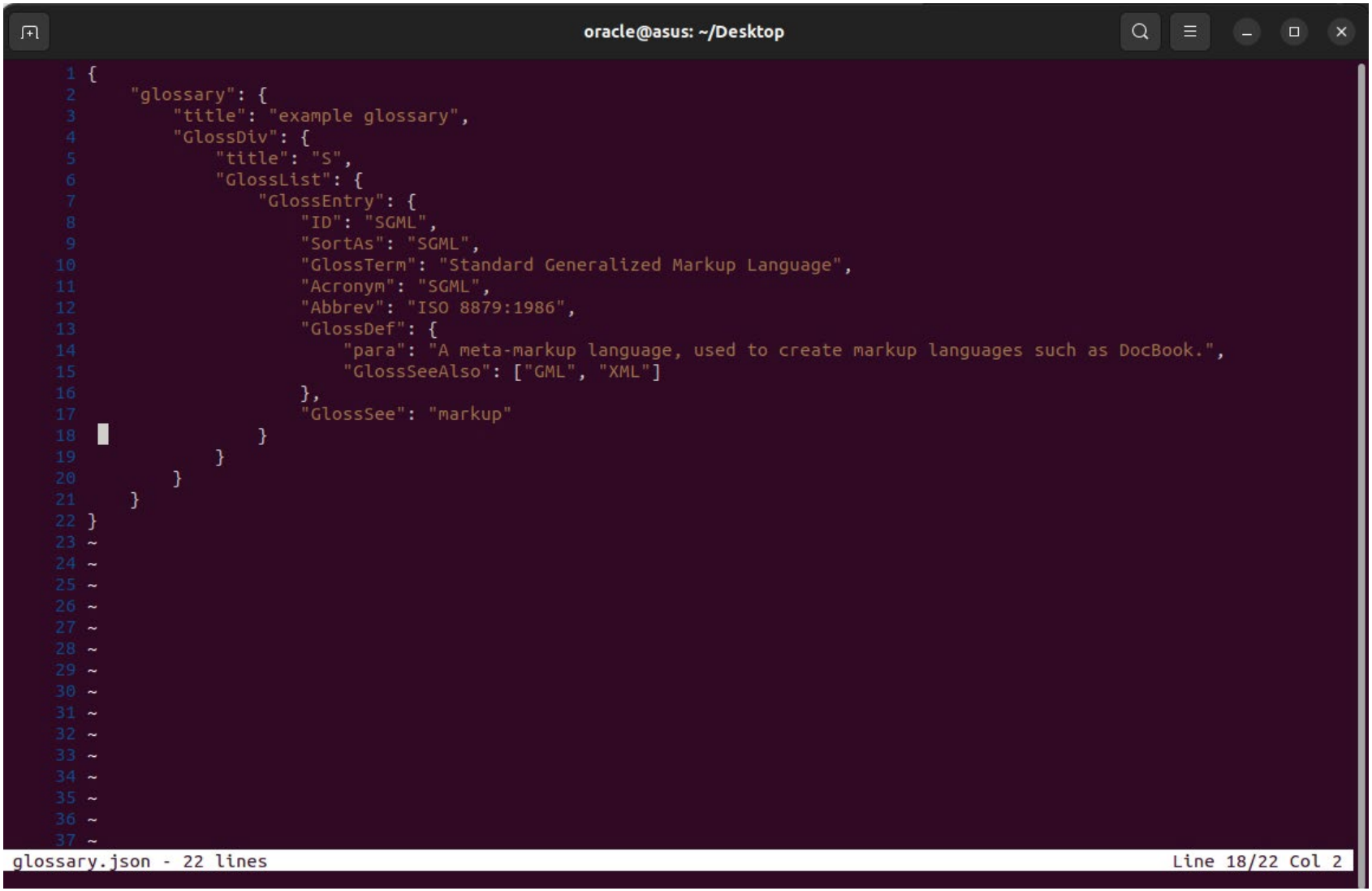
Java language highlighting support.



JavaScript language highlighting support.

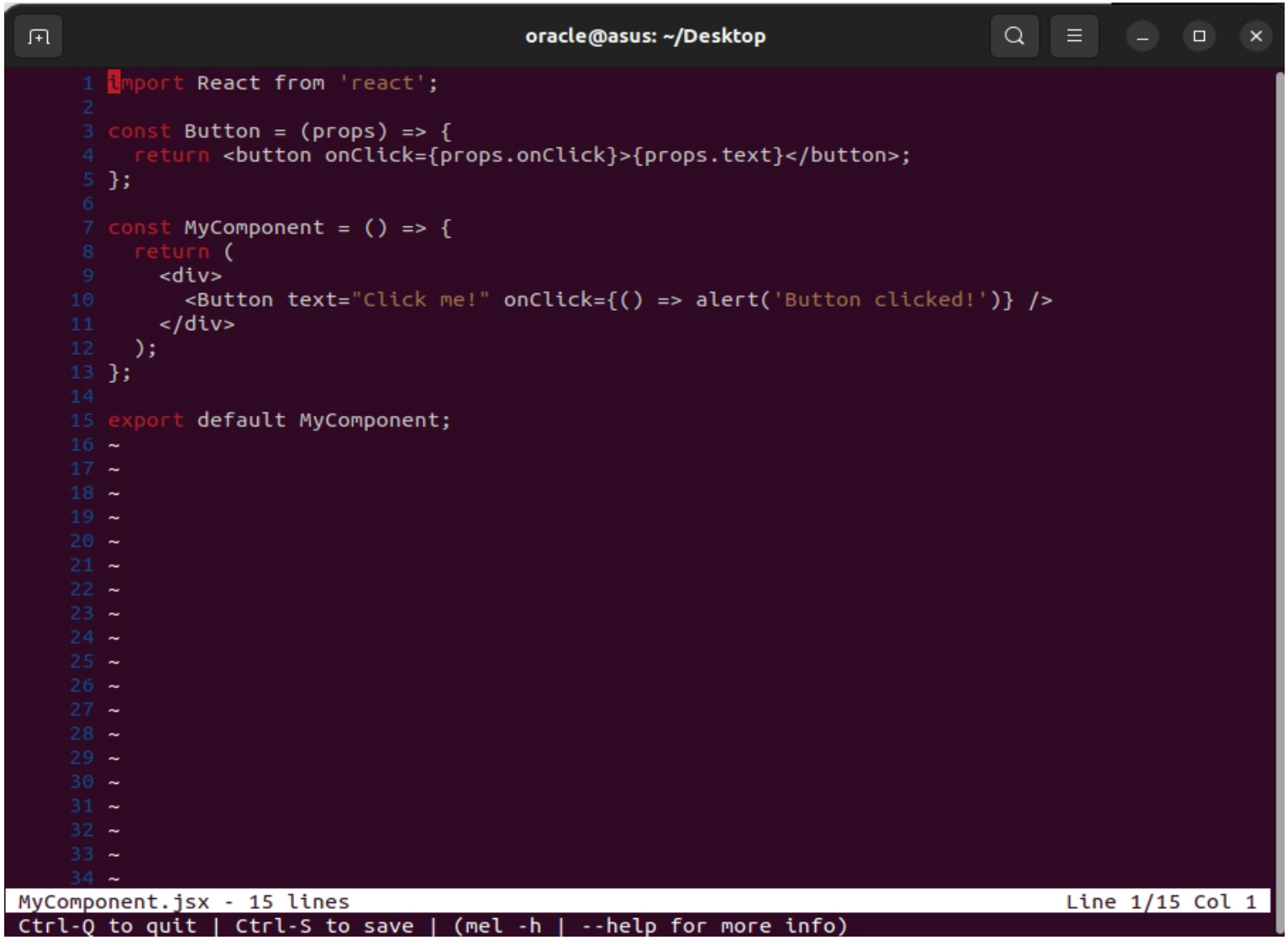


JSON files formatting highlighting support.

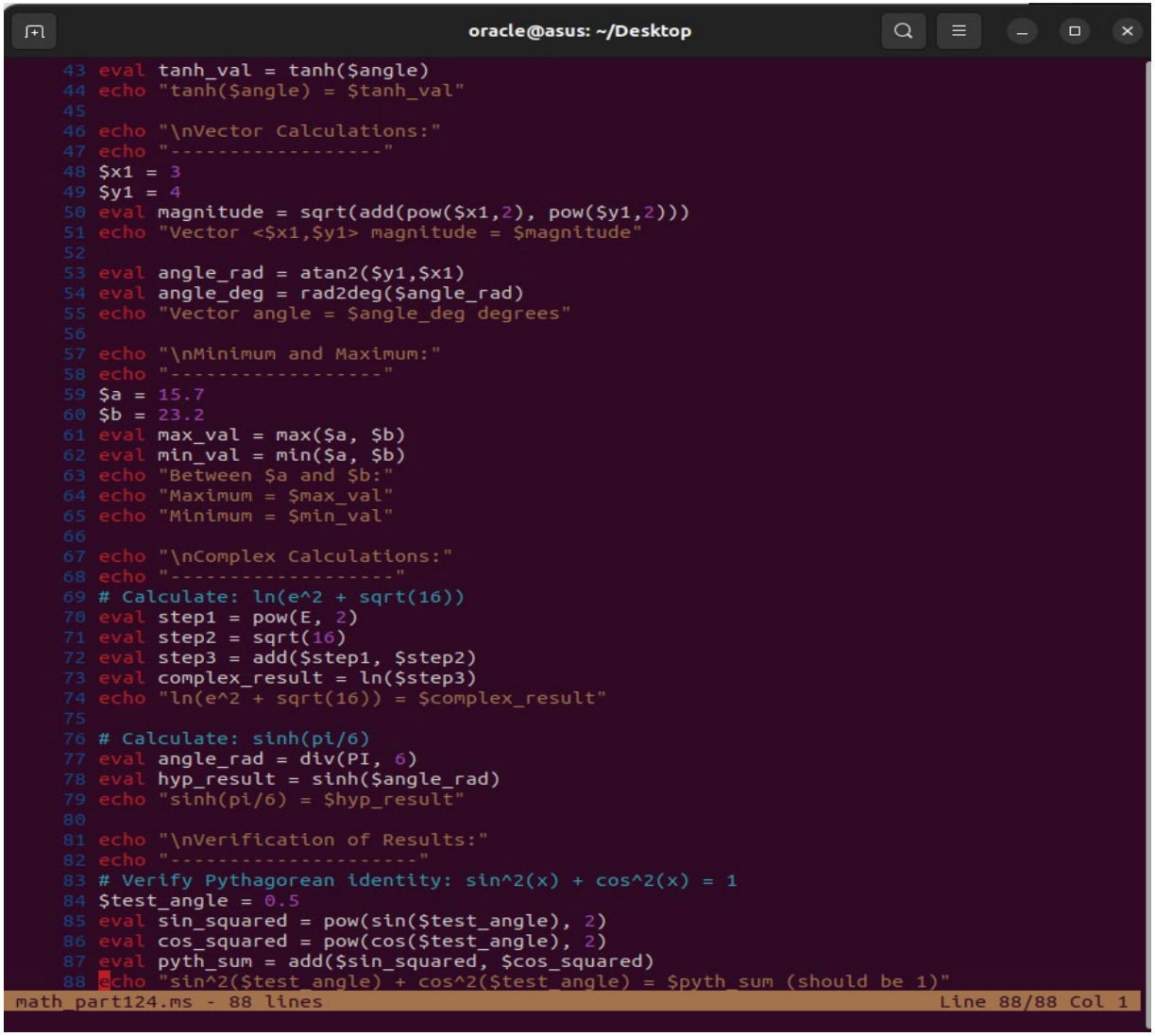


JSONP files formatting highlighting support.

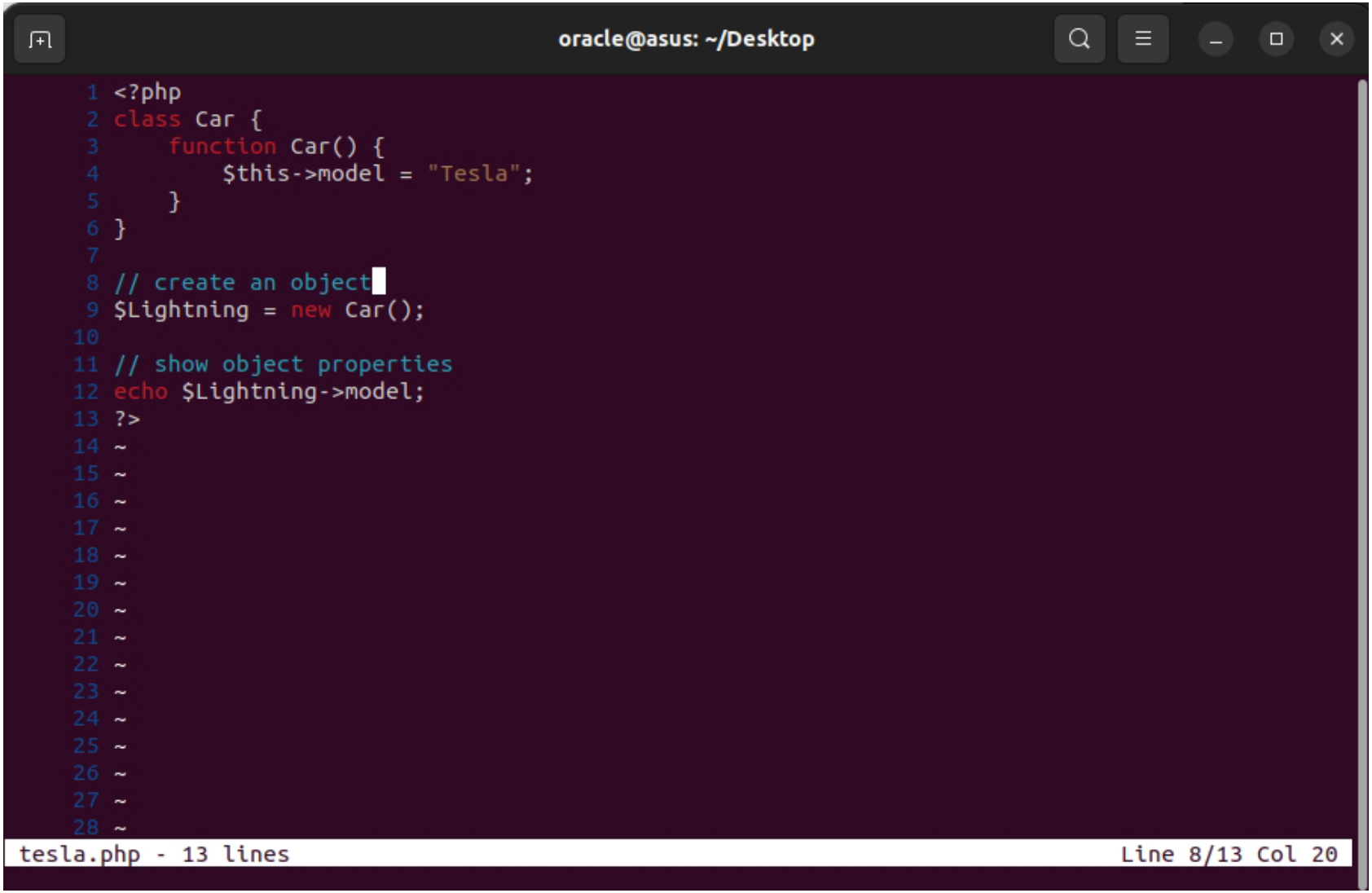
JSX files formatting highlighting support.



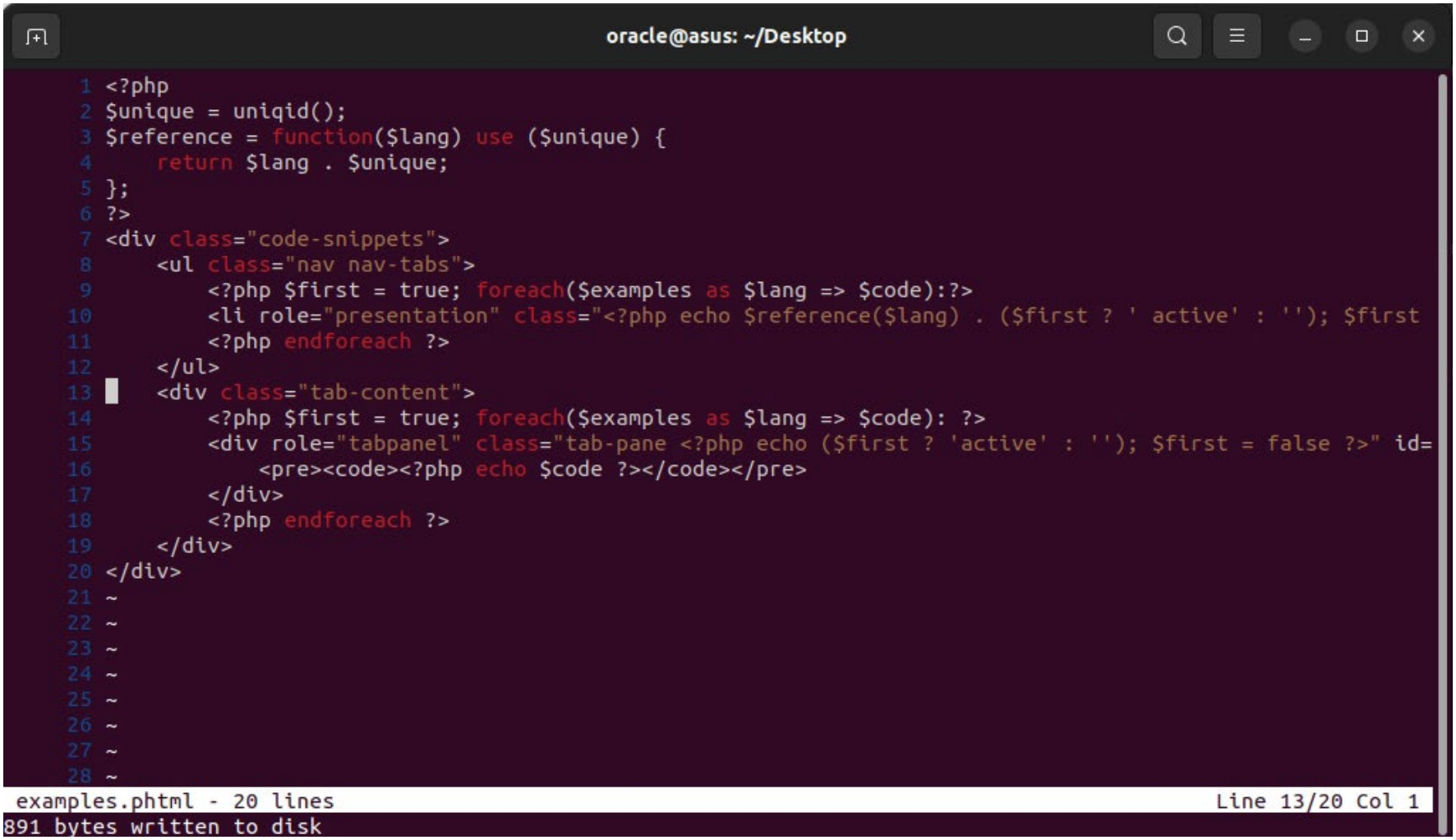
Mshell scripts highlighting support.



Php language highlighting support.



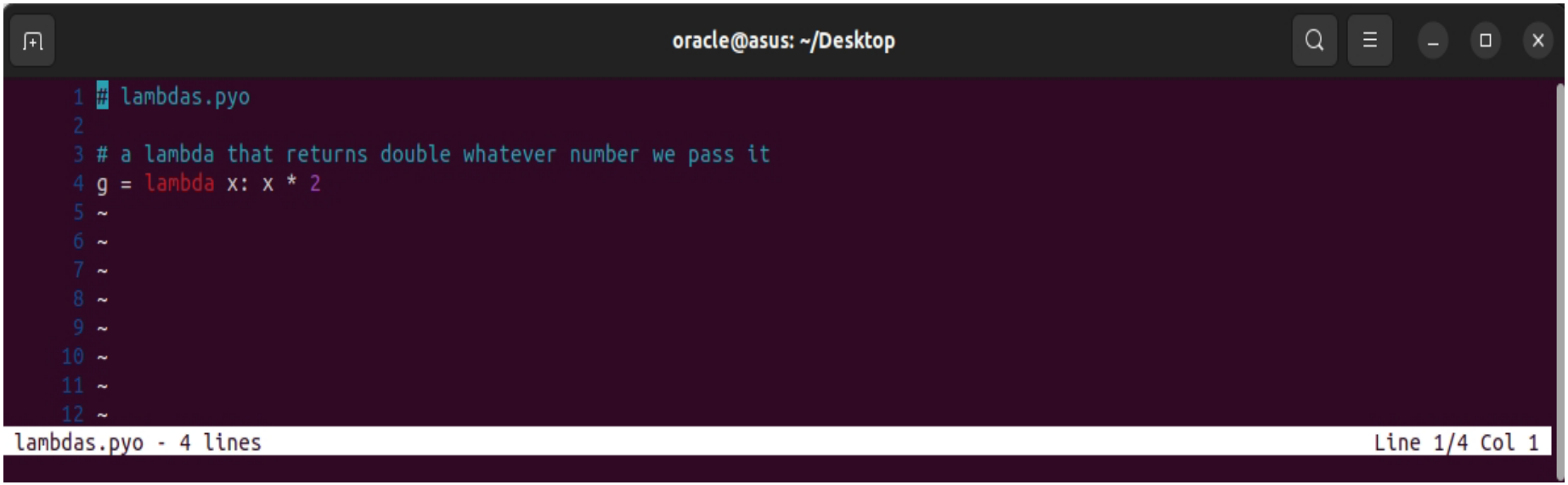
Phtml markup language highlighting support.



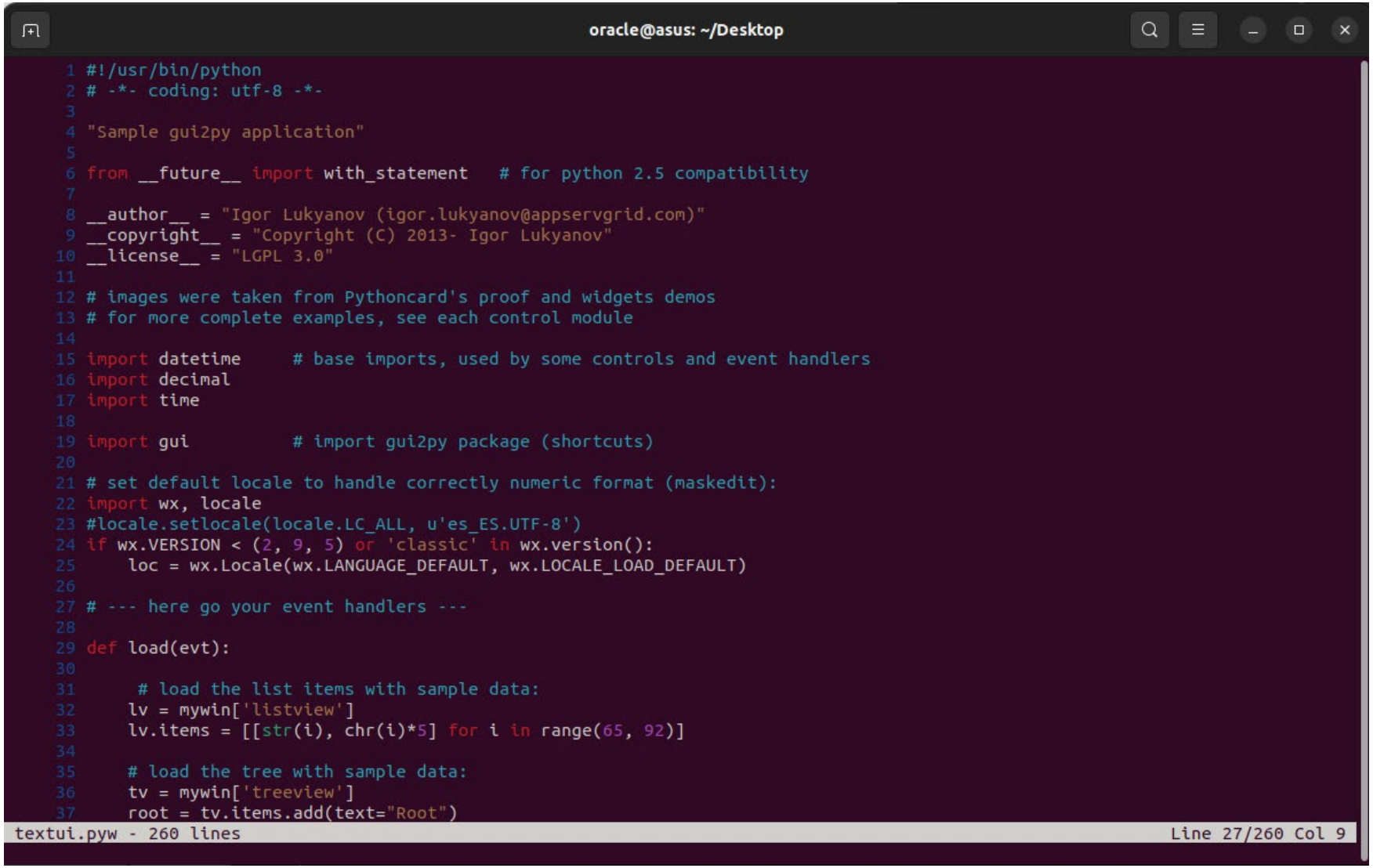
Py3 language files formatting highlighting support.

Pyc language files formatting highlighting support.

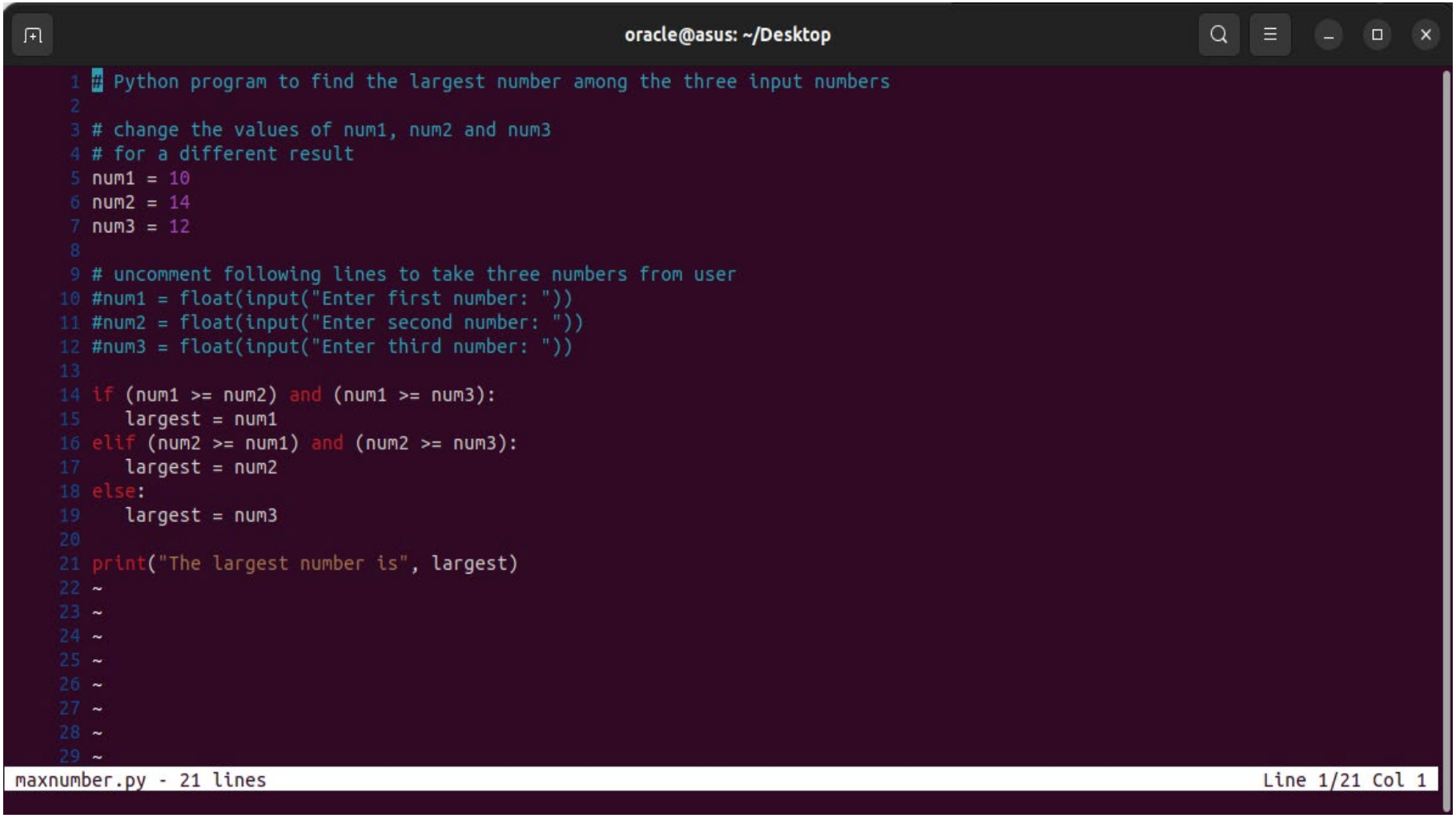
Pyo language files formatting highlighting support.



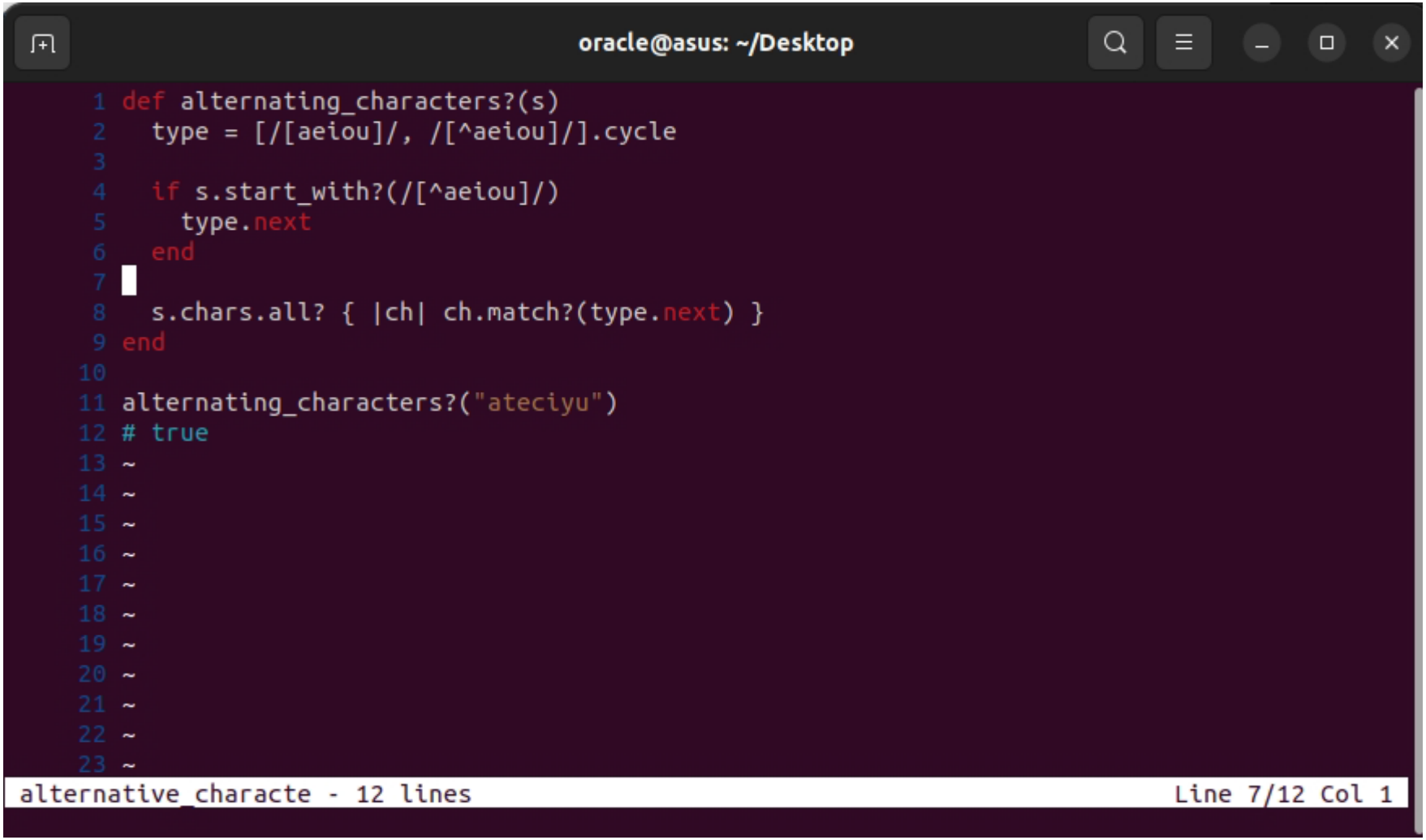
Pyw language files formatting highlighting support.



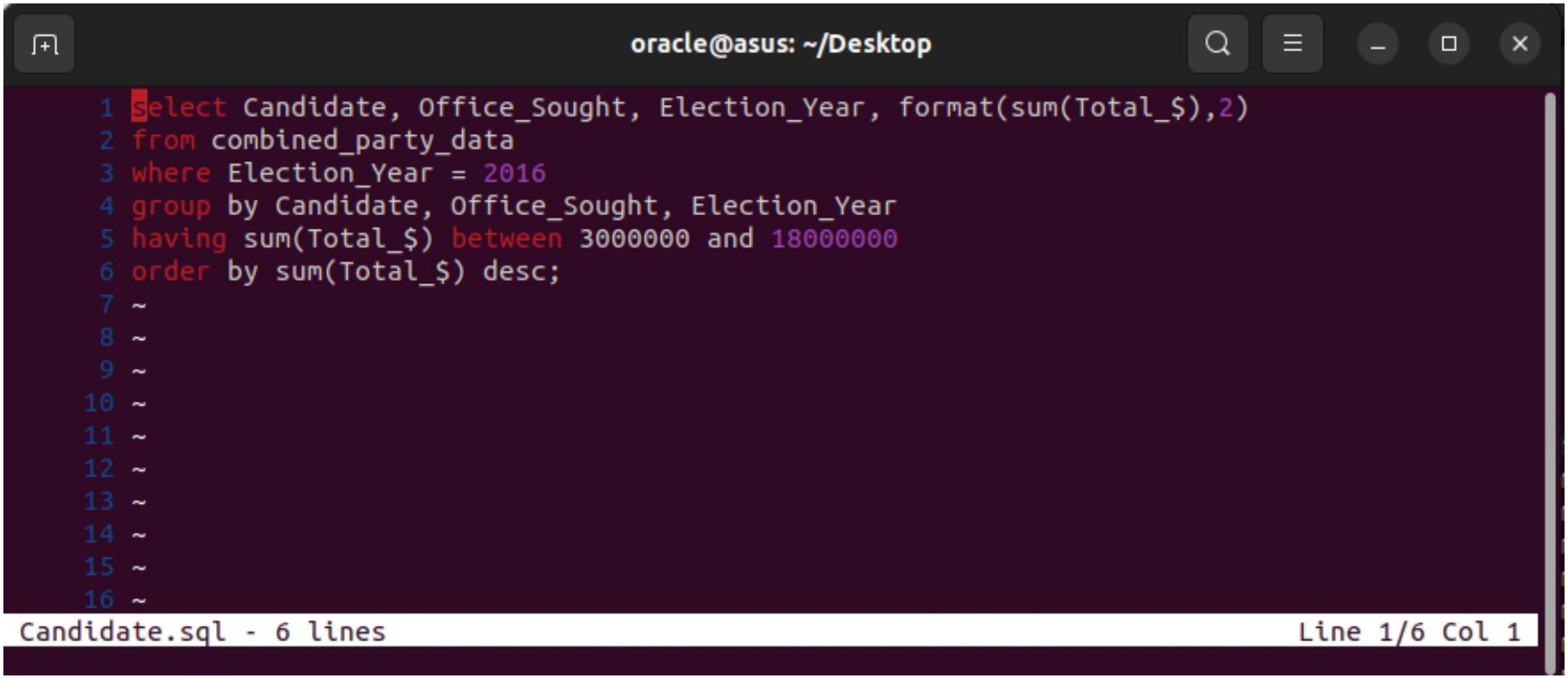
Python language files formatting highlighting support.



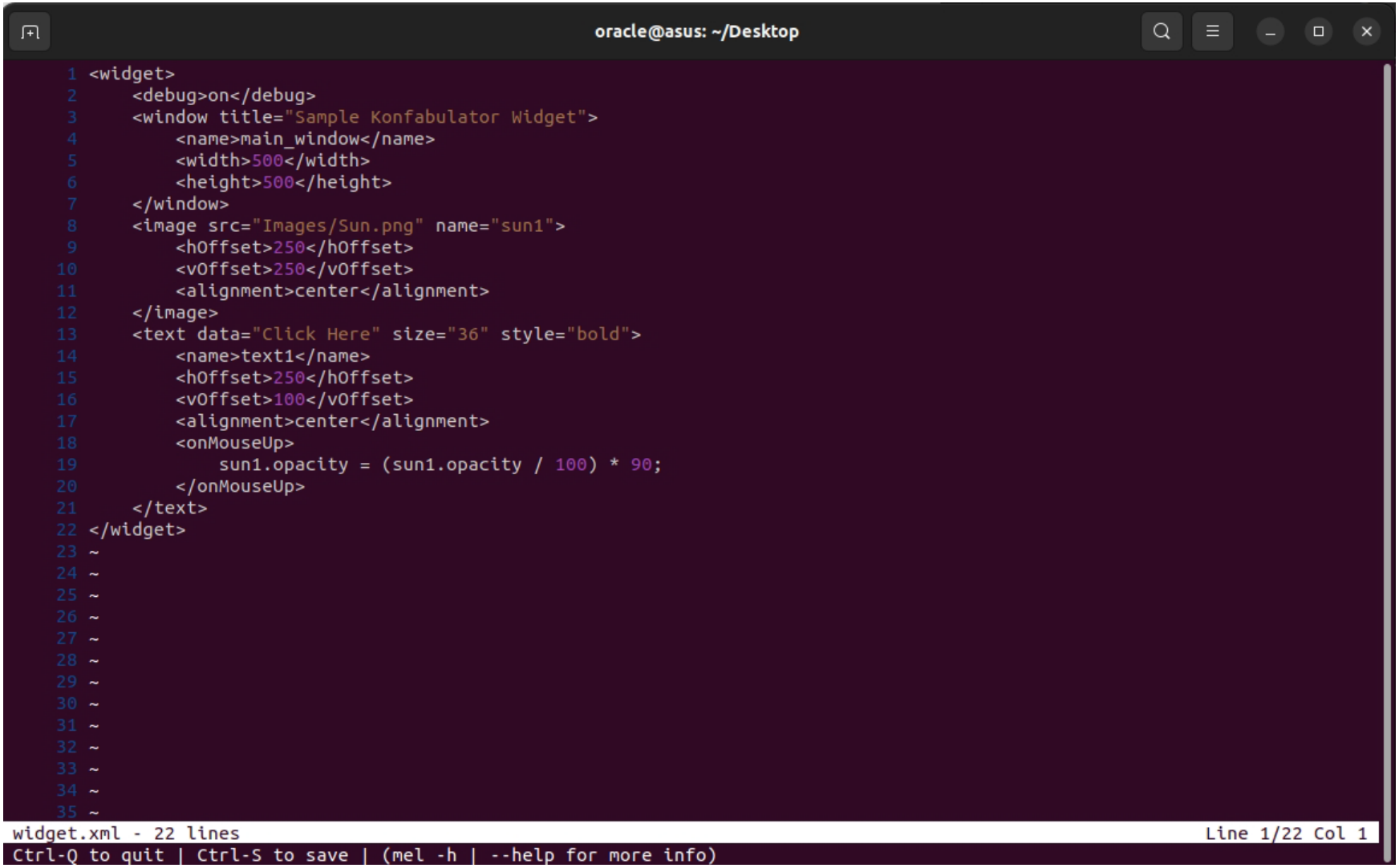
Ruby language files formatting highlighting support.



SQL language files formatting highlighting support.



XML files formatting highlighting support.



CLI options of Mel Editor.

```
OPTIONS
-----------------------------------
Option                                  Action

-h | --help                             Prints the help
-v | --version                          Prints the version of mel
-b | --backup                           Create backup (.bak) file before saving
-l | --line  <number> <file_name>       Open file with cursor on specified line number
-w | --width <columns>                  Set visual column width marker
-----------------------------------
```

CLI options of Mel Editor (‘mel –h’ or ‘mel - -help’).

```
OPTIONS
-----------------------------------
Option                                  Action

-h | --help                             Prints the help
-v | --version                          Prints the version of mel
-b | --backup                           Create backup (.bak) file before saving
-l | --line  <number> <file_name>       Open file with cursor on specified line number
-w | --width <columns>                  Set visual column width marker
-----------------------------------
```

CLI options of Mel Editor (‘mel –h’ or ‘mel - -help’).

```
oracle@asus:~$ mel -h
```

```
oracle@asus:~$ mel -h
Usage: mel [OPTIONS] [FILE]


KEYBINDINGS
-----------

Keybinding    Action

Ctrl-Q        Exit, 3 times click Ctrl-Q if file was changed without saving
Ctrl-S        Save, requires input of file name, if file didn't exist
Ctrl-F        Search by pattern, Esc - exit from Search, Enter and Arrows to interact searching
Ctrl-N        Forward Search by pattern after Ctrl-F. Esc - exit from Search, works after Ctrl-F only
Ctrl-R        Backward Search by pattern after Ctrl-F. Esc - exit from Search, works after Ctrl-F only
Ctrl-J        Global replacement of character combinations, Input Search and Replace patterns, Esc to cancel, Enter to input
Ctrl-G        Go to line Number, requires input the line number
Ctrl-B        Hide/Show line numbering
Ctrl-E        Flip line upwards
Ctrl-D        Flip line downwards
Ctrl-C        Copy line
Ctrl-X        Cut line
Ctrl-V        Paste line
Ctrl-Z        Undo
Ctrl-Y        Redo
Ctrl-P        Pause mel (type "fg" to resume)
Ctrl-W        Retrieve Ollama LLM response
Ctrl-H        Toggle this help screen
Home          Move the cursor to the beginning of the line
End           Move cursor to end of line
PgUp          Up page scroll
PgDn          Down page scroll
Up            Move cursor up one position
Down          Move cursor down one position
Left          Move cursor left one position
Right         Move cursor right one position
Backspace     Delete character


OPTIONS
-----------------------------------
Option                                      Action

-h | --help                                 Prints the help
-v | --version                              Prints the version of mel
-b | --backup                               Create backup (.bak) file before saving
-l | --line  <number> <file_name>           Open file with cursor on specified line number
-w | --width <columns>                      Set visual column width marker
-----------------------------------
Supports highlighting for C,C++,Java,Bash,Mshell,Python,PHP,Javascript,JSON,XML,SQL,Ruby,Go
License: Public domain libre software GPL3,v.0.2.0, 2025
Initial coding: Igor Lukyanov, igor.lukyanov@appservgrid.com
For now, usage of UTF-8 is recommended.
oracle@asus:~$
```

CLI options to check version of Mel Editor (‘mel –v’ or ‘mel - -version’).

```
oracle@asus:~/Desktop$ mel --version
mel - version 0.2.0
oracle@asus:~/Desktop$
```

CLI option to make backup of editing file (‘mel –b name_of_editing_file’ or ‘mel - -backup name_of_editing_file’).

- CLI option to make backup of editing file (‘mel –b name_of_editing_file’ or ‘mel - -backup name_of_editing_file’). Backup of initial editing file is creating after changes and first saving of editing file.
- A backup of the original file is created with the suffix .bak

```
/home/igor/bren3277 > mel -b bren3277.c
```

```
total 112
drwxrwxr-x  2 igor igor  4096 Feb  1 17:54 ./
drwxr-xr-x 50 igor igor  4096 Feb  1 17:50 ../
-rw-rw-r--  1 igor igor 50490 Feb  1 17:54 bren3277.c
-rw-rw-r--  1 igor igor 50489 Feb  1 17:54 bren3277.c.bak
```

CLI option to make backup of editing file (‘mel –b name_of_editing_file’ or ‘mel - -backup name_of_editing_file’).

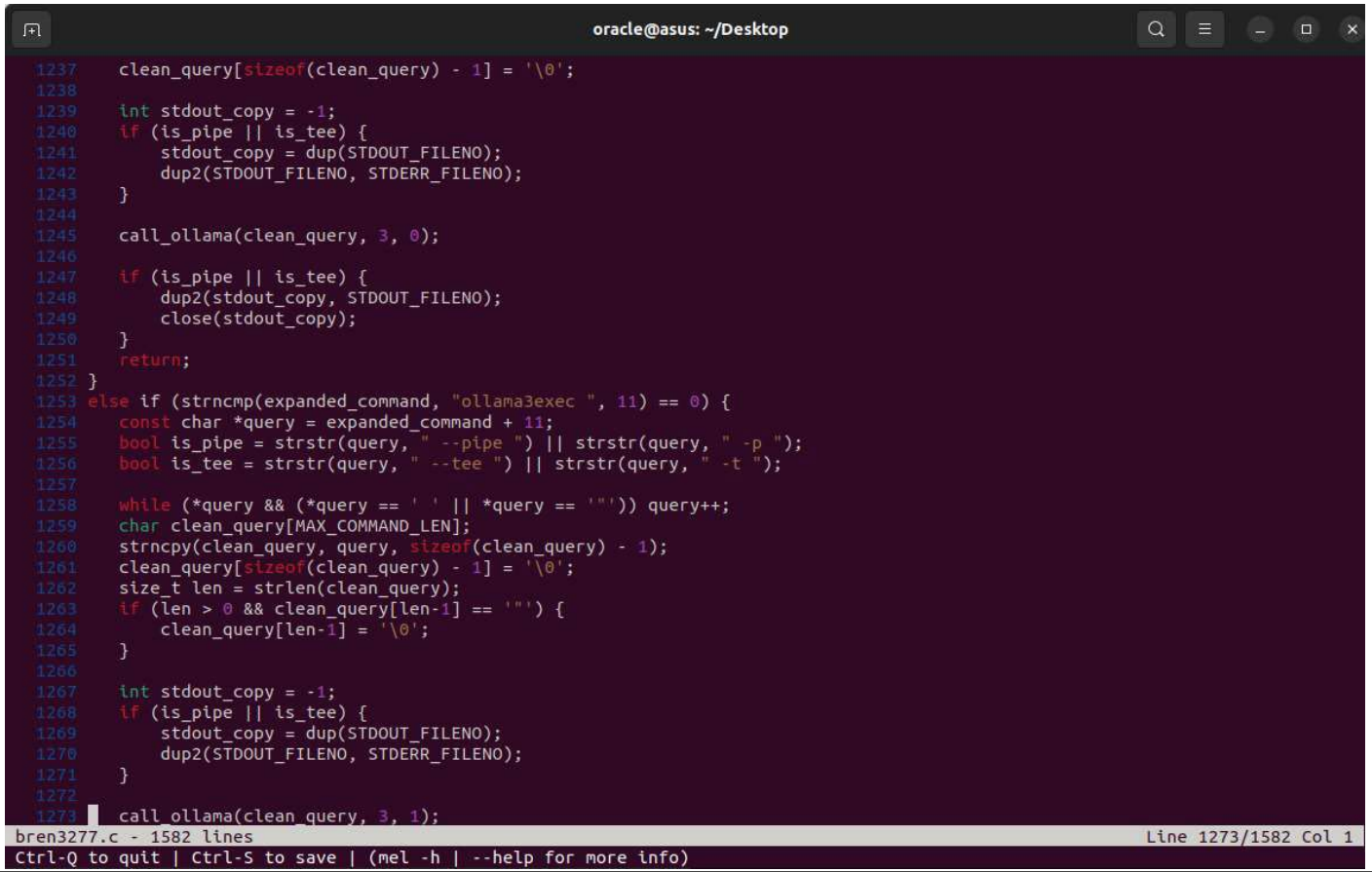
- CLI option to make backup of editing file (‘mel –b name_of_editing_file’ or ‘mel - -backup name_of_editing_file’). Backup of initial editing file is creating after changes and first saving of editing file.
- A backup of the original file is created with the suffix .bak

```
/home/igor/bren3277 > mel -b bren3277.c
```

```
total 112
drwxrwxr-x  2 igor igor  4096 Feb  1 17:54 ./
drwxr-xr-x 50 igor igor  4096 Feb  1 17:50 ../
-rw-rw-r--  1 igor igor 50490 Feb  1 17:54 bren3277.c
-rw-rw-r--  1 igor igor 50489 Feb  1 17:54 bren3277.c.bak
```

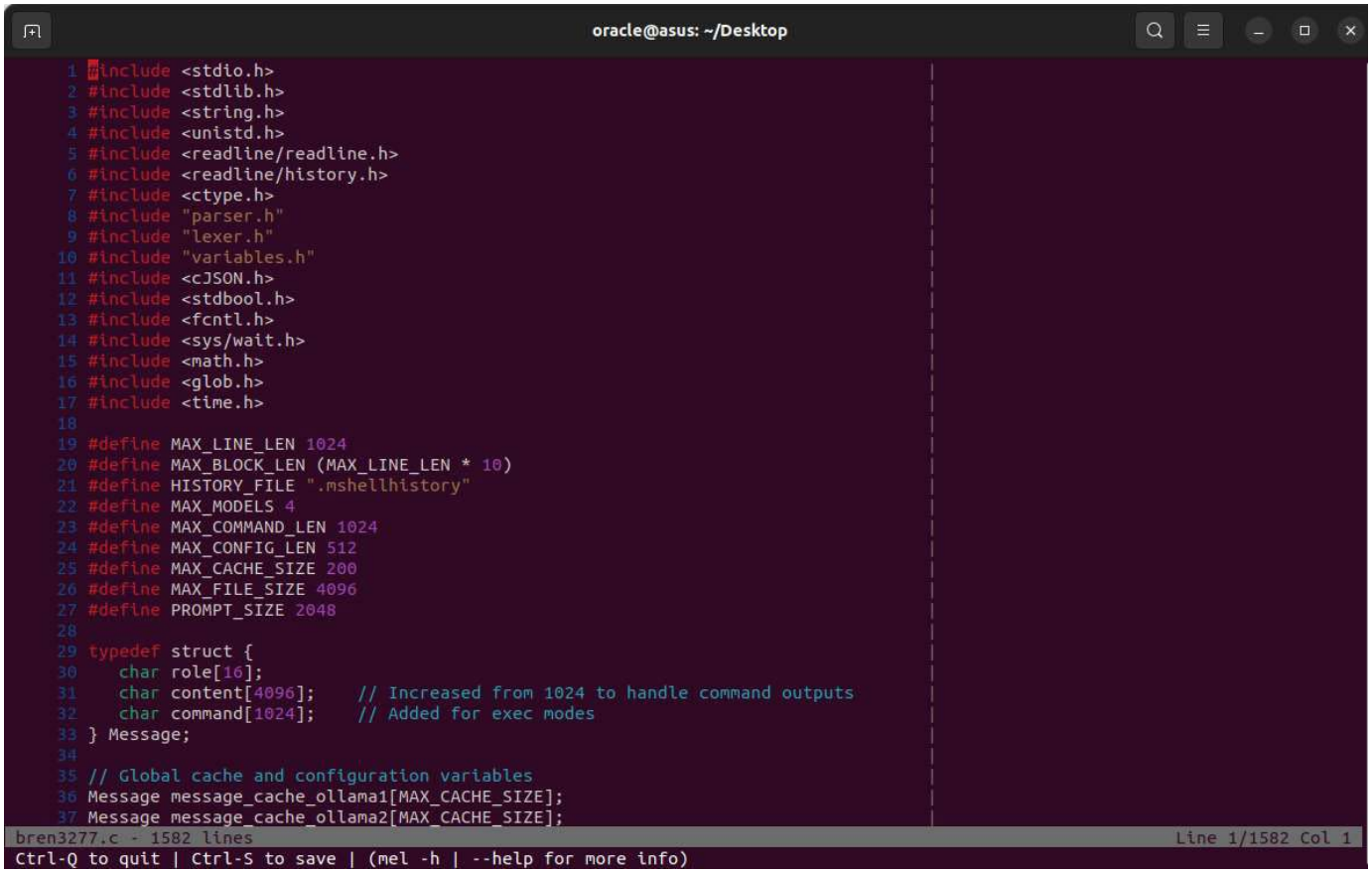
CLI option to open file on specified line numberf (‘mel –l line_number name_of_editing_file’ or ‘mel - -line line_number name_of_editing_file’).

```
/home/igor/bren3277 > mel -l 1273 bren3277.c
```



CLI option to open file on specified line numberf (‘mel –w width_marker_number name_of_editing_file’ or ‘mel - -width width_marker_number name_of_editing_file’).

```
/home/igor/bren3277 > mel -w 82 bren3277.c
```



Multiple CLI options - I.

```
/home/igor/bren3277 > mel -b -l 124 -w 80 bren3277.c
```

```
oracle@asus:~$ mel -b -l 87 aaa.py
```

```
oracle@asus:~$ mel -b -w 80 aaa.py
```

```
oracle@asus:~$ mel -l 45 -w 83 aaa.py
```

Multiple CLI options - II.

```
oracle@asus:~$ mel -b -l 45 -w 83 aaa.py
```

oracle@asus: ~

```python
import os
from pdf2image import convert_from_path
import music21
from PIL import Image
import cv2
import numpy as np

def pdf_to_images(pdf_path, output_folder):
    """
    Convert PDF pages to images.
    """
    images = convert_from_path(pdf_path)
    image_paths = []
    for i, image in enumerate(images):
        output_path = os.path.join(output_folder, f"page_{i + 1}.png")
        image.save(output_path, "PNG")
        image_paths.append(output_path)
    return image_paths

def recognize_music(image_path):
    """
    Recognize musical notation in the image and create a Music21 score.
    (Placeholder function - replace with an actual OMR library like Audiveris)
    """
    # Example placeholder logic: creates a basic C major scale
    score = music21.stream.Stream()
    for note_name in ['C4', 'D4', 'E4', 'F4', 'G4', 'A4', 'B4', 'C5']:
        score.append(music21.note.Note(note_name, quarterLength=1))
    return score

def generate_midi(score, output_midi_path):
    """
    Generate a MIDI file from a Music21 score.
    """
    score.write('midi', fp=output_midi_path)
    print(f"MIDI file created: {output_midi_path}")

def main():
    pdf_path = 'input.pdf'  # Replace with your PDF file path
    output_folder = 'output_images'
    os.makedirs(output_folder, exist_ok=True)

    # Convert PDF to images
    image_paths = pdf_to_images(pdf_path, output_folder)
    print(f"PDF converted to images: {image_paths}")

    # Process each image and generate MIDI
    for i, image_path in enumerate(image_paths):
        print(f"Processing image: {image_path}")
        score = recognize_music(image_path)
```

How to load content for editing (file) – I.

```
/home/igor/mshell/src > mcl mshell.c
```

oracle@asus: ~

```python
import os
from pdf2image import convert_from_path
import music21
from PIL import Image
import cv2
import numpy as np

def pdf_to_images(pdf_path, output_folder):
    """
    Convert PDF pages to images.
    """
    images = convert_from_path(pdf_path)
    image_paths = []
    for i, image in enumerate(images):
        output_path = os.path.join(output_folder, f"page_{i + 1}.png")
        image.save(output_path, "PNG")
        image_paths.append(output_path)
    return image_paths

def recognize_music(image_path):
    """
    Recognize musical notation in the image and create a Music21 score.
    (Placeholder function - replace with an actual OMR library like Audiveris)
    """
    # Example placeholder logic: creates a basic C major scale
    score = music21.stream.Stream()
    for note_name in ['C4', 'D4', 'E4', 'F4', 'G4', 'A4', 'B4', 'C5']:
        score.append(music21.note.Note(note_name, quarterLength=1))
    return score

def generate_midi(score, output_midi_path):
    """
    Generate a MIDI file from a Music21 score.
    """
    score.write('midi', fp=output_midi_path)
    print(f"MIDI file created: {output_midi_path}")

def main():
    pdf_path = 'input.pdf'  # Replace with your PDF file path
    output_folder = 'output_images'
    os.makedirs(output_folder, exist_ok=True)

    # Convert PDF to images
    image_paths = pdf_to_images(pdf_path, output_folder)
    print(f"PDF converted to images: {image_paths}")

    # Process each image and generate MIDI
    for i, image_path in enumerate(image_paths):
        print(f"Processing image: {image_path}")
        score = recognize_music(image_path)
```

How to load content for editing (pipe) – II.

```
/home/igor > cat .bashrc | mel
```

```
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
    *i*) ;;
      *) return;;
esac

# don't put duplicate lines or lines starting with space in the history.
# See bash(1) for more options
HISTCONTROL=ignoreboth

# append to the history file, don't overwrite it
shopt -s histappend

# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000

# check the window size after each command and, if necessary,
# update the values of LINES and COLUMNS.
shopt -s checkwinsize

# If set, the pattern "**" used in a pathname expansion context will
# match all files and zero or more directories and subdirectories.
#shopt -s globstar

# make less more friendly for non-text input files, see lesspipe(1)
[ -x /usr/bin/lesspipe ] && eval "$(SHELL=/bin/sh lesspipe)"

# set variable identifying the chroot you work in (used in the prompt below)
if [ -z "${debian_chroot:-}" ] && [ -r /etc/debian_chroot ]; then
    debian_chroot=$(cat /etc/debian_chroot)
fi

[No Name] - 117 lines                                    Line 1/117 Col 1
Ctrl-Q to quit | Ctrl-S to save | (mel -h | --help for more info)
```

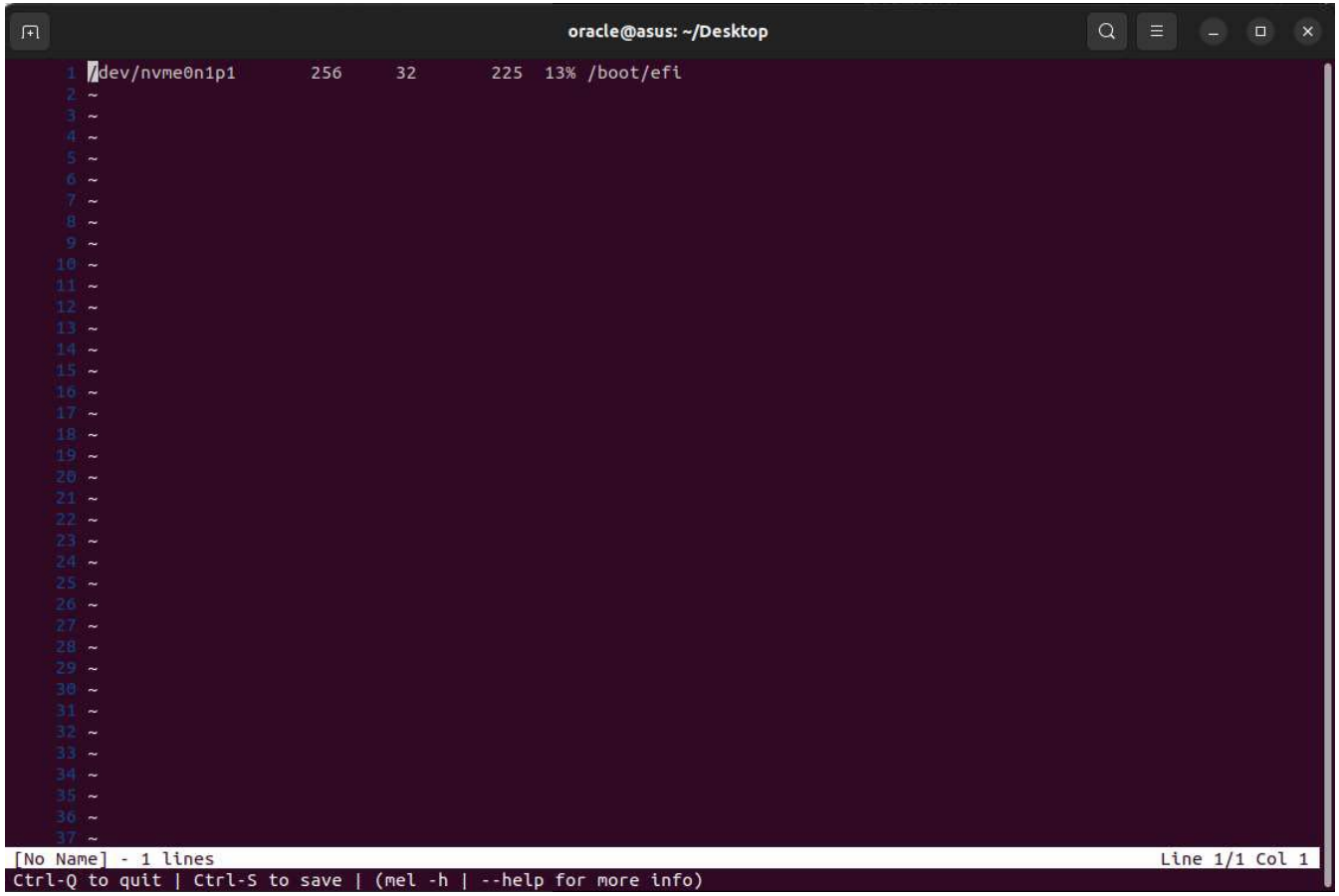
How to load content for editing (commands) – III.

```
/home/igor > (df -m; ls -la; inxi) | mel
```

```
 48 drwxr-xr-x  4 igor   igor     4096 Jan 25 23:09 mshell_backup_January25_2025_fixation
 49 drwxr-xr-x  4 igor   igor     4096 Jan  9 11:57 mshell_backup_January9_2025
 50 drwxr-xr-x  4 igor   igor     4096 Jan 25 16:55 mshell_backup_January9_2025_lua_fixed
 51 drwxrwxr-x  4 igor   igor     4096 Feb  1 00:09 mshell_Feb1_work_on_statements
 52 -rw-r--r--  1 igor   igor    57004 Feb  3 11:59 .mshellhistory
 53 drwxr-xr-x  4 igor   igor     4096 Jan 26 16:37 mshell_jan26_v020
 54 drwxr-xr-x  4 igor   igor     4096 Jan 26 16:38 mshell_jan26_v020_with_debug
 55 drwxr-xr-x  4 igor   igor     4096 Jan 27 04:03 mshell_jan27-fixation
 56 drwxr-xr-x  4 igor   igor     4096 Jan 27 13:19 mshell_jan27_propmpts_path_time_fixation
 57 drwxr-xr-x  4 igor   igor     4096 Jan 27 13:50 mshell_jan27_while_statement_fixation
 58 drwxr-xr-x  4 igor   igor     4096 Jan 27 05:17 mshell_jan27_wildcards_fixation
 59 drwxr-xr-x  4 igor   igor     4096 Jan 22 07:50 mshell_January22_beforecall_ollama_change
 60 drwxr-xr-x  4 igor   igor     4096 Jan 22 22:57 mshell_January23_propbelswithollamapipes
 61 drwxr-xr-x  4 igor   igor     4096 Jan 23 04:12 mshell_January23_propbelswithollamapipes_2
 62 drwxr-xr-x  4 igor   igor     4096 Jan 23 14:34 mshell_January23_propbelswithollamapipes_3
 63 drwxr-xr-x  4 igor   igor     4096 Jan 24 17:48 mshell_January24_lua_fixation
 64 drwxr-xr-x  4 igor   igor     4096 Jan  9 21:22 mshell_January9_9pm
 65 drwxr-xr-x  4 igor   igor     4096 Jan 14 16:06 mshell_ollama1exec
 66 drwxr-xr-x  4 igor   igor     4096 Jan 14 20:12 mshell_ollama1exec2
 67 drwxr-xr-x  4 igor   igor     4096 Jan 17 23:32 mshell_pipesfixation_forinternalcommands_january17_2025
 68 drwxr-xr-x  4 igor   igor     4096 Jan 18 00:24 mshell_pipesfixation_forinternalcommands_withdebug_january17_2025
 69 -rw-r--r--  1 igor   igor      816 Jan 28 22:28 .mshellrc
 70 -rw-r--r--  1 igor   igor      725 Jan 26 02:22 .mshellrc_old
 71 drwxr-xr-x  4 igor   igor     4096 Jan  7 13:30 mshell_real_backup
 72 drwxr-xr-x  3 igor   igor     4096 Jan  6 15:53 mshell_without_lua_support
 73 -rwxr--r--  1 root   root   196865 Jan 30 05:34 mshell.zip
 74 drwxr-xr-x  2 igor   igor     4096 Jan 28 16:27 .ollama
 75 -rw-rw-r--  1 igor   igor     1579 Jan 17 17:27 planets.json
 76 -rw-r--r--  1 igor   igor      807 Dec 30 05:06 .profile
 77 -rw-------  1 igor   igor        7 Jan 23 19:40 .python_history
 78 drwx------  3 igor   igor     4096 Jan 27 01:09 snap
 79 drwx------  2 igor   igor     4096 Feb  1 01:21 .ssh
 80 -rw-r--r--  1 igor   igor        0 Jan  7 17:40 .sudo_as_admin_successful
 81 drwxr-xr-x  2 root   root     4096 Jan 30 04:14 test_highlighting
 82 CPU: quad core Intel Core i5-8265U (-MT MCP-) speed/min/max: 798/400/3900 MHz
 83 Kernel: 6.8.0-52-generic x86_64 Up: 1d 14h 24m Mem: 3294.0/19714.9 MiB (16.7%)
 84 Storage: 6.83 TiB (2.7% used) Procs: 337 Shell: sh inxi: 3.3.13
[No Name] - 84 lines                                          Line 77/84 Col 1
```

How to load content for editing (commands) – IIIa.

```
/home/igor > (df -m | grep /boot/efi) | mel
```



How to load content from llm's with supporting context – IIIb.

```
/home/igor/mel > ollama2 "What makes function time at C language?" | mel
```

oracle@asus: ~/Desktop

```
DEBUG_EXEC: Got command: ollama2 "What makes function time at C language?"
The term "function time" in C programming typically refers to the execution time of a specific function
 or block of code within a program. This can be measured and analyzed to understand performance characteristics,
 optimize code, or identify bottlenecks.

In C, you can measure function execution time using various methods. One common approach is to use high-resolution timers
provided by the operating system or libraries such as `<time.h>` on Unix-like systems. Here's a simple example of how
to measure the execution time of a function:

```c
#include <stdio.h>
#include <time.h>

// Function whose execution time we want to measure
void myFunction() {
    // Simulate some computations
    for (int i = 0; i < 1000000; i++) {
        int j = i * i;
    }
}

int main() {
    clock_t start, end;
    double cpu_time_used;

    start = clock(); // Record the start time

    myFunction(); // Call the function whose execution time we want to measure

    end = clock(); // Record the end time

    cpu_time_used = ((double) (end - start)) / CLOCKS_PER_SEC; // Calculate the ela
DEBUG_PIPE: Child 1: Command completed
~
~
~
~
```

123.txt - 33 lines Line 8/33 Col 1

1198 bytes written to disk
```

Installation from binaries.

```
KEYBINDINGS
-----------

Keybinding    Action

Ctrl-Q        Exit, 3 times click Ctrl-Q if file was changed without saving
Ctrl-S        Save, requires input of file name, if file didn't exist
Ctrl-F        Search by pattern, Esc - exit from Search, Enter and Arrows to interact searching
Ctrl-N        Forward Search by pattern after Ctrl-F. Esc - exit from Search, works after Ctrl-F only
Ctrl-R        Backward Search by pattern after Ctrl-F. Esc - exit from Search, works after Ctrl-F only
Ctrl-J        Global replacement of character combinations, Input Search and Replace patterns, Esc to cancel, Enter to input
Ctrl-G        Go to line Number, requires input the line number
Ctrl-B        Hide/Show line numbering
Ctrl-E        Flip line upwards
Ctrl-D        Flip line downwards
Ctrl-C        Copy line
Ctrl-X        Cut line
Ctrl-V        Paste line
Ctrl-Z        Undo
Ctrl-Y        Redo
Ctrl-P        Pause mel (type "fg" to resume)
Ctrl-W        Retrieve Ollama LLM response
Ctrl-H        Toggle this help screen
Home          Move the cursor to the beginning of the line
End           Move cursor to end of line
PgUp          Up page scroll
PgDn          Down page scroll
Up            Move cursor up one position
Down          Move cursor down one position
Left          Move cursor left one position
Right         Move cursor right one position
Backspace     Delete character
```

Keybindings.

```
KEYBINDINGS
-----------

Keybinding    Action

Ctrl-Q        Exit, 3 times click Ctrl-Q if file was changed without saving
Ctrl-S        Save, requires input of file name, if file didn't exist
Ctrl-F        Search by pattern, Esc - exit from Search, Enter and Arrows to interact searching
Ctrl-N        Forward Search by pattern after Ctrl-F. Esc - exit from Search, works after Ctrl-F only
Ctrl-R        Backward Search by pattern after Ctrl-F. Esc - exit from Search, works after Ctrl-F only
Ctrl-J        Global replacement of character combinations, Input Search and Replace patterns, Esc to cancel, Enter to input
Ctrl-G        Go to line Number, requires input the line number
Ctrl-B        Hide/Show line numbering
Ctrl-E        Flip line upwards
Ctrl-D        Flip line downwards
Ctrl-C        Copy line
Ctrl-X        Cut line
Ctrl-V        Paste line
Ctrl-Z        Undo
Ctrl-Y        Redo
Ctrl-P        Pause mel (type "fg" to resume)
Ctrl-W        Retrieve Ollama LLM response
Ctrl-H        Toggle this help screen
Home          Move the cursor to the beginning of the line
End           Move cursor to end of line
PgUp          Up page scroll
PgDn          Down page scroll
Up            Move cursor up one position
Down          Move cursor down one position
Left          Move cursor left one position
Right         Move cursor right one position
Backspace     Delete character
```

License.

```
/*
 *
 *
 *    This program is free software: you can redistribute it and/or modify
 *    it under the terms of the GNU General Public License as published by
 *    the Free Software Foundation, either version 3 of the License, or
 *    any later version.
 *
 *    This program is distributed in the hope that it will be useful,
 *    but WITHOUT ANY WARRANTY; without even the implied warranty of
 *    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 *    GNU General Public License for more details.
 *
 *    You should have received a copy of the GNU General Public License
 *    along with this program.  If not, see <http://www.gnu.org/licenses/>.
 */
```

Questions

&

Answers

Contact information:

Igor Lukyanov, Art2Dec SoftLab

- **E-mail support for MEL editor: igor101964@gmail.com**
- **NIC-handle: LIK26**
- **WWW: http://www.appservgrid.com**
- **E-resume: http://www.art2dec.co/paw1/**

Art2Dec®