Lahim by Michał Nowakowski

# PRZEMYŚLENIA

Projekt w Języku Skryptowym 2023

Spis treści:



https://github.com/nowakowski-m/pjs_michal_nowakowski_2023

# Czego się nauczyłem.

Dzięki temu projektowi i błędom jakie występowały podczas jego tworzenia nauczyłem się sporo o samym Linuxie, trochę też o Pythonie.

Była to fajna lekcja, bo jednak na co dzień nie korzystam z bardzo wielu plików, rzadko też mają np. ograniczone uprawnienia, albo jakieś inne specyficzne cechy, które okazuje się mają swoje „ukryte" przedstawienia w systemach UNIX, np.:

- Ukryte pliki – ich nazwa rozpoczyna się od kropki (.)
- Rozszerzenia plików – nie są w ogóle znaczące, są opcjonalne i używane głównie do identyfikacji zawartości przez użytkownika.
- Uprawnienia – każdy plik ma przypisane uprawnienia dla użytkowników lub grup, dotyczy to możliwości odczytu, zapisu, oraz wykonania pliku.

Tutaj zaś musiałem przeanalizować takie wypadki, co pozwoliło mi na zdobycie wiedzy, chociażby o powyżej wymienionych przykładach.

Co do Pythona, na pewno było tu najwięcej korzystania z Debuggera ile kiedykolwiek mi się zdarzyło. Analizowanie działania aplikacji krok po kroku jest ważną umiejętnością, pozwalającą nie tylko wyeliminować znalezione już błędy, ale też odnaleźć nowe nieprawidłowości. Czasami są to niuanse, które nie powodują chociażby wyłączenia aplikacji, ale mogą prowadzić do nieoptymalnego działania, lokowania zbyt dużo pamięci, itp.

# Spostrzeżenia dot. Bibliotek.

Zauważyłem też że na miejscu twórców biblioteki Curses która posłużyła mi do stworzenia interfejsu w terminalu, inaczej rozwiązałbym komunikaty błędów, które użytkownik otrzymuje gdy aplikacja się "wysypuje".

Są one zdecydowanie zbyt ogólne. Np. częstym błędem jaki otrzymywałem było:

"Curses -> Addstr returned err" in line ...

A więc dawało mi to jedynie wiedze, że błąd tkwi w funkcji nakładającej tekst na ekran, użytej w danej linijce kodu. Czysto teoretycznie pisząc to teraz, wydaje mi się że brzmi to nieźle i mogłoby się wydawać, że jest tam wszystko czego potrzeba programiście.

W rzeczywistości jednak, bardzo często miałem problem z późniejszą diagnozą problemu, a to chociażby dlatego, że przykładowe błędy jakie potem wychodziły, wynikały z zbyt wielu linijek do dodania na ekranie, funkcja po prostu dawała error gdy próbowaliśmy dodać tekst w miejscu gdzie nie ma już ekranu, lub też gdy tekst był za długi w danej linijce.

Robił się z tego spory problem, ponieważ nawet do końca nie wiedziałem jak na bieżąco wyglądają listy plików generowane przez program, a co za tym idzie nie wiedziałem, co próbował dodać na ekran i dlaczego się to mu nie udało.

# Praca nad projektem.

Po za "kodową" częścią projektu, bardzo fajnie było stworzyć sobie jakiś plan działania, ustalić harmonogramy i czuć delikatną presję co do jego utrzymywania.

Podobało mi się, że nasze projekty są mniej więcej na bieżąco sprawdzane i otrzymujemy jakieś opinie.

Wszystko wymagało zawarcia w prezentacjach, które przedstawialiśmy przed publiką. To też było pozytywne doświadczenie, bo pozwalało nauczyć się jak przedstawiać plany, problemy, itd. Dodatkowo można było zainteresować inne osoby swoim projektem, co powodowało późniejsze dyskusje w gronie o technologiach, zamysłach, itp.

Myślę, że jest to jakiś zarys tego co może nas kiedyś czekać w pracy, jeśli się zdecydujemy na zostanie programistą w jakimś Teamie.

Przy okazji poćwiczyliśmy korzystanie z GitHuba oraz jak ktoś to robił, również z samego Gita.

Wiele konkretów dotyczących zmian zostało zawartych w commitach na Githubie, gdzie zainteresowani mogą zobaczyć finalny kod, jego proces powstawania oraz modyfikacji. Każdy z nich ma odpowiedni opis, który pozwala łatwiej znaleźć interesującą nas zmianę.