

# Comparative analysis of package managers Flatpak and Snap used for open-source software distribution

## Analiza porównawcza menedżerów pakietów Flatpak i Snap wykorzystywanych do dystrybucji oprogramowania o otwartym kodzie

Grzegorz Jan Cichocki\*, Sławomir Wojciech Przyłucki

*Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland*

### Abstract

This article presents the result of a research of the Flatpak and Snap package managers used to distribute open-source software on Linux systems. Both package managers are characterised by their versatility and implementation of sandboxing. As part of the research, a test application was prepared, which was built in the Flatpak and Snap formats and published in the official software repositories, where for Flatpak it is Flathub and for Snap it is the Snap Store. The prepared application was first used to test and compare the implementation of sandboxing rules. This was followed by tests of RAM usage and start-up time by the application installed in both formats. The result of the research is an analysis of the measurement results and the drawing of conclusions.

*Keywords:* Flatpak; Snap; package manager; open-source; application performance

### Streszczenie

Artykuł przedstawia wynik badań nad menedżerami pakietów Flatpak oraz Snap wykorzystywanych do dystrybucji oprogramowania o otwartym kodzie w systemach Linux. Obydwa menedżery pakietów cechują się uniwersalnością oraz implementacją systemu piaskownicy (ang. sandboxing). W ramach badań przygotowana została aplikacja testowa, którą zbudowano w formatach Flatpak i Snap, a także opublikowano w oficjalnych repozytoriach z oprogramowaniem, gdzie dla Flatpak jest to Flathub, a dla Snap – Snap Store. Przygotowaną aplikację najpierw wykorzystano do zbadania i porównania implementacji zasad piaskownicy. Następnie przeprowadzono testy użycia pamięci RAM oraz czasów uruchamiania przez aplikację zainstalowaną w obydwóch formatach. Wynikiem badań jest analiza uzyskanych pomiarów oraz wyciągnięcie wniosków.

*Słowa kluczowe:* Flatpak; Snap; menedżer pakietów; otwarty kod; wydajność aplikacji

\*Corresponding author

Email address: [grzegorz\\_cichocki@protonmail.com](mailto:grzegorz_cichocki@protonmail.com) (G. J. Cichocki)

©Published under Creative Common License (CC BY-SA v4.0)

## 1. Wstęp

W dziedzinie dystrybucji oprogramowania o otwartym kodzie, menedżery pakietów odgrywają kluczową rolę w zapewnieniu efektywnego i bezpiecznego sposobu dostarczania aplikacji do użytkowników. W ostatnich latach dwie technologie, tj. Flatpak i Snap, wyłoniły się jako znaczące rozwiązania w tym obszarze. Obie te platformy przyniosły nowatorskie rozwiązania w zakresie pakowania i dystrybucji aplikacji, a ich popularność nadal rośnie wśród twórców oprogramowania oraz użytkowników systemów Linux.

Podczas badań skupiono się na kilku kluczowych zagadnieniach. Przeanalizowano implementację zasad piaskownicy (ang. sandboxing), która jest istotnym aspektem w kontekście bezpieczeństwa i izolacji aplikacji. Następnie przeprowadzono testy wydajności, w postaci pomiaru użycia pamięci RAM i czasu uruchamiania aplikacji w obu formatach. Wyniki tych badań dają szersze spojrzenie na Flatpaka i Snapa pod kątem efektywności i wydajności.

Niniejszy artykuł stanowi rezultat przeprowadzonych badań, których celem jest dogłębna analiza porównawcza menedżerów pakietów Flatpak i Snap w kontekście dystrybucji oprogramowania open-source.

Skoncentrowano się na kluczowych aspektach technicznych, efektywności i bezpieczeństwa. Opracowana analiza opiera się na konkretnych badaniach, w trakcie których przygotowano aplikację testową o nazwie „Tabela”, zbudowaną zarówno w formacie Flatpak, jak i Snap. Następnie opublikowano tę aplikację w oficjalnych repozytoriach, tj. Flathub dla Flatpaka [1] oraz Snap Store dla Snapa [2].

Aplikacja Tabela jest prostym narzędziem pozwalającym wyświetlić zawartość wybranego pliku CSV w formie graficznych wierszy i kolumn. Dodatkowo, zaznaczane są puste komórki, a także istnieje możliwość dodania kolejnych wierszy. Użytkownik może również zapisać dane do formatu CSV lub XML. Aplikacja została napisana z wykorzystaniem języka programowania Python oraz bibliotek graficznych Qt w wersji 6 (udostępnione dla języka Python jako biblioteka PySide6). Kod źródłowy aplikacji został udostępniony w repozytorium Codeberg [3]. W celu publikacji aplikacji w Snap Store oraz Flathub zostały przygotowane pliki manifestu. Repozytorium Github [4] zawiera plik manifestu dla sklepu z aplikacjami Flathub, a repozytorium Codeberg [5] plik manifestu napisany z myślą o sklepie z aplikacjami Snap Store.

### 1.1. Menedżer pakietów

Menedżer pakietów (ang. Package Manager) to oprogramowanie zazwyczaj wykorzystywane w systemach Linux do instalacji, aktualizacji i usuwania oprogramowania, na przykład aplikacji tekstowych, z interfejsem graficznym czy dodatkowych bibliotek (ang. libraries) oraz innych modułów. Instalacja oprogramowania z wykorzystaniem menedżera pakietów jest procesem koherentnym, to znaczy, że za każdym razem wygląda podobnie. Oprogramowanie jest pobierane z zaufanego serwera zwanego repozytorium i jest ono zazwyczaj już w wersji skompilowanej pod architekturę procesora komputera wykorzystywanego przez użytkownika. Wraz z plikami binarnymi potrzebnymi do instalacji programu, pobierane są inne metadane, które zawierają, między innymi, opis aplikacji, informacje o autorach, licencji i inne dane identyfikacyjne, a także dodatkowe, wymagane zależności oraz skrypty instalacyjne. Menedżery również często sprawdzają skróty kryptograficzne, na przykład sumy kontrolne SHA-256, aby upewnić się o autentyczności i zgodności ze źródłem pobranych pakietów. Dodatkowo, niektóre menedżery pakietów mogą również wykorzystywać podpisy cyfrowe do weryfikacji.

### 1.2. Flatpak

Flatpak to jeden z najpopularniejszych uniwersalnych menedżerów pakietów. Może być on wykorzystywany w wielu dystrybucjach i działa na warstwie systemowej i aplikacji. Menedżer pakietów wywodzi się z projektu „xdg-app” opracowywanego w 2014 roku. Flatpak bazuje na pracach wykonanych przez programistę Alexandra Larssona, a w projekt zainwestowało wiele znanych firm, w tym Red Hat, Endless Computers oraz Collabora.

Użytkownicy mogą zarządzać aplikacjami Flatpak na różne sposoby, zarówno za pomocą interfejsów graficznych, jak i tekstowych. Ważnym narzędziem jest aplikacja Flatseal pozwalająca zarządzać uprawnieniami. Aplikacje Flatpak są dystrybuowane za pomocą repozytoriów, gdzie Flathub jawi się jako największy katalog, zawierający ponad 2800 programów. Dostępne są zarówno aplikacje o otwartym kodzie, jak i własnościowe. Można także instalować aplikacje za pomocą plików .flatpak i .flatpakref.

Cechą wyróżniającą Flatpak jest pakowanie aplikacji w kontenery, które zawierają wszelkie niezbędne zależności. Podczas pracy, aplikacja znajduje się w izolowanym środowisku, tak zwanej piaskownicy, która zazwyczaj zawiera program oraz środowisko uruchomieniowe. Aplikacja działa nad warstwą systemu, dzięki czemu może być uruchamiana w wielu dystrybucjach.

Flatpak wykorzystuje mechanizm portali (ang. portals), co pozwala skonteneryzowanym aplikacjom komunikować się z innymi usługami systemowymi. Mowa tu, między innymi, o systemie plików, czy dostępie do mikrofonu. Kolejnym, ważnym składnikiem aplikacji Flatpak jest środowisko uruchomieniowe (ang. runtime). Każda aplikacja musi być zbudowana w oparciu o środowisko uruchomieniowe. Środowiska uruchomie-

niowe zapewniają również podstawowe składniki wykorzystywane przez programy. W jednym systemie można zainstalować wiele środowisk uruchomieniowych, w tym w wielu wersjach. Środowiska uruchomieniowe są niezależne od dystrybucji i poszczególnych wersji systemów operacyjnych.

Ze względu na to, że Flatpak nie wykorzystuje dostępnych w systemie bibliotek, menedżer pobiera wszelkie zależności ponownie. Aby ograniczyć zajętość dyskową, wprowadzono mechanizm deduplikacji środowisk uruchomieniowych i innych składników. Flatpak w pewnym stopniu jest w stanie współdzielić pewne moduły, aby obniżyć zajętość dyskową, a deduplikacja ma tym bardziej pozytywny wpływ, im więcej aplikacji Flatpak zainstalowano w systemie.

### 1.3. Snap

Snap to uniwersalny menedżer pakietów stworzony przez firmę Canonical, zaprojektowany dla systemów Linux. Pierwotnie, Snap był dostępny tylko w dedykowanej dystrybucji Ubuntu Core, jednak w 2016 roku został rozszerzony na systemy desktopowe.

Użytkownik może zainstalować oprogramowanie Snap na różne sposoby, w tym poprzez Snap Store dostępny w systemach Ubuntu. Oprogramowanie w Snap Store pochodzi z centralnego repozytorium, dostarczanego zarówno przez niezależnych twórców, jak i firmy komercyjne. W Snap Store dostępne jest oprogramowanie o otwartym kodzie oraz własnościowe.

Moduł dystrybucji oprogramowania w Snap jest kontrowersyjny, ponieważ choć sama technologia Snap jest otwarta, tak Snap Store pozostaje jedynym repozytorium zdalnym, nad którym kontrolę ma wyłącznie firma Canonical, a kod źródłowy repozytorium nie jest dostępny publicznie. Menedżer pakietów Snap został zoptymalizowany tylko pod kątem Snap Store oraz dystrybucji z wykorzystaniem lokalnych plików .snap.

Snap bazuje na systemie inicjalizacji Systemd dostępnym w większości dystrybucji. Jest to moduł potrzebny, ponieważ w systemie operacyjnym wykorzystującym menedżer Snap uruchamiany jest proces działający w tle (ang. daemon) o nazwie „snapd”. Jego zadaniem jest ciągła i dynamiczna obsługa aplikacji oraz procesów zainstalowanych za pomocą tego menedżera pakietów. Aplikacje Snap są również automatycznie aktualizowane w tle.

Snap wykorzystuje mechanizmy izolacji. Zainstalowana aplikacja działa w specjalnym kontenerze, co znacznie zwiększa bezpieczeństwo i prywatność użytkownika. W celu dostępu do wybranych elementów, na przykład urządzeń USB, mikrofonu czy kamery wideo, aplikacje Snap wykorzystują moduł o nazwie „Interfejsy” (ang. Interfaces). Interfejsy Snap często wykorzystują domyślne interfejsy API systemów Linux oraz standaryzowane interfejsy XDG Desktop Portals. Użytkownik może dowolnie zarządzać uprawnieniami danych aplikacji do poszczególnych elementów. Modyfikacja uprawnień jest możliwa z wykorzystaniem aplikacji Snap Store oraz interfejsu tekstowego w konsoli. Programy Snap są dystrybuowane razem ze wszelkimi

zależnościami potrzebnymi do działania oprogramowania, tym samym nie są zależne od składników systemowych.

Snap wykorzystuje również system plików SquashFS. Jest to system plików tylko do odczytu (ang. read-only filesystem) stosowany w systemach operacyjnych, który został zaprojektowany do efektywnego kompresowania i przechowywania danych. Aplikacje Snap są zwykle opakowane w pojedyncze pliki SquashFS o rozszerzeniu „.snap”, które zawierają wszystkie pliki, biblioteki i zależności wymagane do działania aplikacji. Metadane zawarte w pliku SquashFS są interpretowane przez snapd w celu skonfigurowania odpowiednio ukształtowanej bezpiecznej piaskownicy dla aplikacji. Po zainstalowaniu aplikacji Snap, w systemie użytkownika plik jest montowany i dekompresowany wtedy, kiedy to potrzebne. W systemie plików pliki Snap są widziane jako abstrakcyjne urządzenia w formie „loop”.

Unikatową zaletą uniwersalnego menedżera pakietów Snap jest możliwość dostarczania różnych typów aplikacji, a także całych stosów oprogramowania. W repozytorium Snap Store dostępne są nie tylko aplikacje z interfejsem graficznym czy narzędzia konsolowe, ale także pełne aplikacje serwerowe zawierające wiele składników, w tym serwer WWW, silnik bazodanowy, interfejs użytkownika oraz wykonywalny kod.

## 2. Przegląd literatury

### 2.1. Menedżery pakietów

Wynikiem prac nad rozwiązaniem, które pomogłyby w zarządzaniu pakietami, jak i aplikacjami oraz ich zależnościami, są menedżery pakietów. Jak zauważa autor publikacji [6] nie tylko instalacja oprogramowania stała się łatwiejsza, ale też ich aktualizacja oraz utrzymanie. O wiele łatwiej mają też deweloperzy, twórcy oprogramowania. Publikacja aplikacji jest prostsza, a użytkownik ma ułatwioną instalację – również poprzez oprogramowanie z interfejsem graficznym GUI (ang. Graphical User Interface). Wokół menedżerów pakietów zbudowano wiele aplikacji, które za pomocą kilku kliknięć myszą pozwalają na instalację określonego oprogramowania. Największą zaletą wykorzystywania menedżerów pakietów jest ponowne wykorzystywanie komponentów zewnętrznych (ang. third-party) bez ponoszenia konsekwencji w postaci przywołanego wcześniej zjawiska „Dependency Hell” oraz w formie lepszego wykrywania problemów z kompatybilnością jeszcze na etapie produkcji oprogramowania.

Autorzy pracy [7] również podkreślają znaczenie nowoczesnych menedżerów pakietów jako niezwykle ważnego składnika w procesie dystrybucji oprogramowania o otwartym kodzie, jednak także zauważają problemy wynikające z ich architektury. Jak zaznaczają, aktualnie wykorzystywane menedżery pakietów cechują się budową monolityczną i doraźnymi mechanizmami rozwiązywania problemów z zależnościami oraz niestandardowymi heurystykami. Zdaniem autorów, rozwiązaniem dostrzeżonych problemów mogą być modułowe menedżery pakietów.

W pracy [8] skupiono się na bezpieczeństwie menedżerów pakietów. W ramach badań przeprowadzono audyt bezpieczeństwa menedżera pakietów CPAN (ang. Comprehensive Perl Archive Network) dla języka Perl. Autorzy również zbadali popularne menedżery pakietów pod kątem szerokiej gamy wytycznych dotyczących bezpieczeństwa. Badanie integracji wykazało, że niektóre z menedżerów pakietów nie wykorzystują nawet tak prostej metody zabezpieczającej jak sprawdzanie sum kontrolnych danych udostępnionych oraz danych pobranych. Jak wynika z badań, menedżer cabal-install nie sprawdza sum kontrolnych, a EasyInstall tylko wtedy, kiedy takową sumę kontrolną dostarczy autor oprogramowania. Dla kontrastu, CPAN zawsze bada sumy kontrolne.

### 2.2. Uniwersalne menedżery pakietów

Standardowe menedżery pakietów w systemach Linux często są powiązane z konkretnymi dystrybucjami. Dystrybucja Debian i bazująca na niej dystrybucja Ubuntu wykorzystuje APT (ang. Advanced Packaging Tool), a dystrybucja Fedora oferuje DNF (ang. Dandified YUM). Autor pracy [9] przedstawia problematykę standardowych menedżerów pakietów. Rozważane są takie zagadnienia jak przywiązanie danego menedżera do jednej dystrybucji, co utrudnia dystrybucję oprogramowania, a także o utrudnionym dostępie do nowszych wersji aplikacji – często dana wersja wybranej aplikacji jest ściśle powiązana z wersją systemu operacyjnego.

W omawianym artykule wskazano, że uniwersalne menedżery pakietów są rozwiązaniem opisanych problemów. Autor skupia się na menedżerze Flatpak, który jest dostępny w wielu dystrybucjach rodziny systemów Linux, a więc jednocześnie można wykorzystywać go na wielu różnych systemach.

Uniwersalne menedżery pakietów nie zawsze są pozytywnie oceniane. Popularnym artykułem jest [10], gdzie autor wskazuje powody, przez które nie uważa menedżera pakietów Flatpak za dobre rozwiązanie dystrybucji aplikacji w systemach typu Linux. Wymienia, między innymi, potrzebę pobierania wielu megabajtów danych często dla prostych aplikacji. Za przykład podaje aplikację KCalc, gdzie sama aplikacja zajmuje 4,4 MB, jednakże wraz z nią użytkownik musi pobrać niemalże 900 MB innych modułów potrzebnych do uruchomienia. W publikacji wyszczególniono także problem z błędnie działającym systemem dzielenia środowisk wykonawczych (ang. runtime) dla aplikacji, a także zwiększone użycie pamięci operacyjnej RAM.

W odpowiedzi na artykuł [10] została przygotowana publikacja [11], w której skonfrontowano argumenty. Problem pobierania wielkich zasobów danych wytłumaczono potrzebą zachowania zgodności i pewności, że dane oprogramowanie będzie bezproblemowo działać. Użytkownicy uruchamiają dany program z takimi składnikami, które określił deweloper. Ma to również pozytywny wpływ na deweloperów, którzy nie będą musieli rozwiązywać wielu problemów zgłaszanych przez użytkowników, które wynikałyby właśnie z błęd-

nych wersji bibliotek czy środowisk wykonawczych (lub ich braku) u użytkownika.

Artykuł porusza także problem dużej zajętości dyskowej przez aplikacje i składniki zainstalowane poprzez menedżer Flatpak. Wskazano, że istnieją mechanizmy w tym menedżerze, których zadaniem jest zaoszczędzenie przestrzeni dyskowej, między wspomniana wcześniej deduplikacja środowisk wykonawczych. Zaznaczono również, że systemy z rodziny Linux, w których to Flatpak jest głównym menedżerem pakietów, często wykazują się mniejszą zajętością dyskową zaraz po instalacji, ponieważ nie zawierają innych modułów, które byłyby potrzebne do zarządzania, instalacji i uruchamiania aplikacji dostarczonych za pomocą tradycyjnych menedżerów pakietów.

### 2.3. Zasada piaskownicy

W publikacji [12] autorzy skupili się na badaniach zasad piaskownicy w systemach z rodziny Linux. Jako dwa prężnie rozwijane menedżery pakietów, które implementują funkcjonalność piaskownicy, wymieniono Flatpak oraz Snap. Autorzy zwracają również uwagę na fakt, że źródło takiej architektury bezpieczeństwa i prywatności ma korzenie w aplikacjach mobilnych, to znaczy aplikacji przeznaczonych na smartfony oraz tablety. Autorzy publikacji podkreślają, że transformacja zasad piaskownicy zależnych od użytkownika (lub konta użytkownika) na zasady oparte na samej aplikacji jest korzystne i wyraźnie widoczne w systemach typu Linux.

W ramach przeprowadzonych badań sprawdzono implementację zasad poprzez zbadanie 283 aplikacji dostępnych w obydwóch formatach, odpowiednio: Flatpak i Snap. Jak wynika z tych badań, aż 90,1% sprawdzonych aplikacji Snap implementuje zasady piaskownicy, natomiast w przypadku menedżera Flatpak jest to tylko 58,3%. Ponadto badacze zaznaczają, że autorzy oprogramowania niekiedy błędnie opisują zasady i uprawnienia swoich produktów. Bywa tak, że dokładnie ta sama aplikacja potrzebowała więcej uprawnień początkowych do działania w jednym menedżerze niż w przypadku drugiego menedżera, przy czym na obydwóch platformach działała ona poprawnie. Autorzy zaznaczają, że postęp w implementacji zasad piaskownicy jest coraz większy, ale nadal wymagane jest wiele usprawnień.

## 3. Eksperyment badawczy

### 3.1. Metody badań

Celem pracy jest przeprowadzenie analizy porównawczej dla dwóch uniwersalnych menedżerów pakietów w systemie Linux, czyli Flatpak i Snap. Porównanie skupia się na wydajności aplikacji okienkowej zainstalowanej w oparciu o obie metody dystrybucji oprogramowania. Zostaną porównane także zasady piaskownicy dostępne w obu menedżerach pakietów w oparciu o aplikację testową. Badania zostaną podzielone na scenariusze badawcze opisane w dalszych podrozdziałach i zostaną przeprowadzone w sposób sekwencyjny z zachowaniem odpowiedniego czasu na ustabilizowa-

nie się temperatury komputera przeznaczonego do wykonania testów. Otrzymane wyniki zostaną szczegółowo omówione i zaprezentowane w formie tabel, wykresów oraz z wykonaniem podstawowej analizy statystycznej.

### 3.2. Scenariusze badań

#### 3.2.1. Scenariusz S1 – Porównanie zasad piaskownicy aplikacji na podstawie przygotowanej aplikacji badawczej o nazwie Tabela.

W tym scenariuszu zostanie przeprowadzona analiza porównawcza zasad piaskownicy aplikacji okienkowych w systemie Linux z wykorzystaniem menedżerów pakietów Flatpak i Snap. Do badania zostaną wykorzystane aplikacje: Flatseal (w przypadku menedżera Flatpak) oraz Snap Store (dla zbadania cech menedżera Snap), w których zostanie zliczona liczba zasad, którymi użytkownik programu może zarządzać. Porównanie zostanie przeprowadzone na podstawie oficjalnie wydanej aplikacji w oparciu o obydwie metody dystrybucji oprogramowania.

Aplikacja Flatseal to dedykowany program towarzyszący do obsługi aplikacji typu Flatpak. Za jego pomocą użytkownik jest w stanie zarządzać uprawnieniami programów. Aplikacja Snap Store pełni rolę przede wszystkim sklepu z oprogramowaniem dystrybuowanym za pomocą menedżera pakietów Snap. Jednakże ma ona też dodatkowe funkcje, w tym kontrolę uprawnień.

#### 3.2.2. Scenariusz S2 – Zbadanie użycia pamięci operacyjnej RAM przez aplikację Tabela zainstalowaną za pomocą Flatpak i Snap

W tym scenariuszu badawczym zostanie zbadane użycie pamięci RAM aplikacji Tabela, która została zainstalowana za pomocą Flatpak oraz Snap. Użycie pamięci zostanie zmierzone wielokrotnie. Pierwsza seria badań uwzględni aplikację Tabela chwilę po uruchomieniu. Następnie zostanie wykonana druga seria badań użycia pamięci RAM przez aplikację Tabela z otworzonym przykładowym plikiem CSV.

Użycie pamięci RAM to jeden z najważniejszych czynników wpływających na wydajność aplikacji okienkowych. Badanie pozwoli wyłonić menedżer pakietów, którego sposób działania pozwala na mniejsze wykorzystanie pamięci operacyjnej. Badanie zostanie wykonane z użyciem narzędzia Sysstat.

#### 3.2.3. Scenariusz S3 – Badanie czasu uruchomienia aplikacji Tabela zainstalowanej z wykorzystaniem metod dystrybucji Flatpak oraz Snap

W tym scenariuszu badawczym zostanie zbadany czas uruchomienia aplikacji Tabela, która została zainstalowana za pomocą Flatpak oraz Snap. Badanie zostanie przeprowadzone za pomocą narzędzia Hyperfine. Test przewiduje zbadanie czasu uruchomienia aplikacji po raz pierwszy od uruchomienia systemu, tak zwany zimny start (ang. cold start) oraz po wcześniejszym uruchomieniu i zamknięciu bez wyłączania systemu, co nazywane będzie gorącym startem (ang. warm start).

Czas uruchomienia to jeden z ważniejszych czynników wpływających na doświadczenia z korzystania z aplikacji okienkowych w systemie operacyjnym. Badając ten parametr można będzie wskazać ten menedżer pakietów, który został lepiej zoptymalizowany pod tym kątem.

### 3.3. Platforma testowa

Do wykonania badań został wykorzystany komputer typu laptop o podanej specyfikacji.

- Procesor: Intel Core i7-1165G7.
- Układ graficzny: Intel Iris Xe.
- Pamięć operacyjna RAM: 16 GB LPDDR4X.
- Dysk: Toshiba KXG60ZNV1T02 NVMe PCI-E 3.0 o pojemności 1 TB.
- System operacyjny: Ubuntu 23.04.

## 4. Wyniki badań

### 4.1. Wyniki dla scenariusza S1

Tabela 1 zawiera porównanie zasad piaskownicy menedżerów pakietów Flatpak i Snap na podstawie aplikacji Tabela. W tabeli wprowadzono podział na kategorie przedstawiony przez aplikację Flatseal. Są to kolejno: podsystemy udostępniane, gniazda piaskownicy, urządzenia, dodatkowe zezwolenia oraz system plików.

Tabela 1: Porównanie zasad piaskownicy menedżerów pakietów Flatpak i Snap na podstawie aplikacji Tabela

Flatpak	Snap
<ul style="list-style-type: none"> <li>• Dostęp do sieci.</li> <li>• Umożliwienie komunikacji między procesami.</li> </ul>	<ul style="list-style-type: none"> <li>• Umożliwienie działania jako serwis sieciowy.</li> </ul>
<ul style="list-style-type: none"> <li>• Wykorzystanie systemu wyświetlania X11.</li> <li>• Wykorzystanie systemu wyświetlania Wayland.</li> <li>• Możliwość wykorzystania systemu wyświetlania X11, jeśli istnieją problemy z Wayland.</li> <li>• Wykorzystanie serwera dźwięku Pulse-Audio</li> <li>• Dostęp do D-Bus sesji.</li> <li>• Dostęp do D-Bus systemu.</li> <li>• Dostęp do SSH.</li> <li>• Dostęp do karty mikroprocesorowej.</li> <li>• Dostęp do obsługi drukowania.</li> <li>• Dostęp do katalogu agenta GPG.</li> </ul>	
<ul style="list-style-type: none"> <li>• Możliwość wykorzystania GPU do przyspieszenia sprzętowego.</li> <li>• Dostęp do wirtualizacji KVM.</li> <li>• Dostęp do współdzielonej pamięci.</li> <li>• Dostęp do urządzeń komputera.</li> </ul>	<ul style="list-style-type: none"> <li>• Dostęp do stosu technologicznego OpenGL.</li> </ul>
<ul style="list-style-type: none"> <li>• Dostęp do programów</li> </ul>	

<ul style="list-style-type: none"> <li>• Dostęp do magistrali CAN.</li> <li>• Dostęp do pamięci współdzielonej dla aplikacji.</li> </ul>	
<ul style="list-style-type: none"> <li>• Możliwość precyzyjnego ustalenia dostępu do plików, w tym do konfiguracji systemowej, bibliotek systemowych, plików wykonywalnych lub danych statycznych, katalogu domowego użytkownika lub do ręcznie wybranego katalogu.</li> <li>• Dostęp do opcji działania w tle.</li> <li>• Możliwość wyświetlania powiadomień.</li> </ul>	<ul style="list-style-type: none"> <li>• Umożliwienie dostępu aplikacji do katalogu domowego użytkownika.</li> </ul>

W przypadku Snap dodatkowe możliwości konfiguracji mogą zostać wyświetlone za pomocą polecenia „snap connections tabela”. Pomijając pozycje przedstawione w Tabeli 1, są to: desktop, desktop-legacy, wayland oraz x11.

Aplikacja Tabela w wersji Flatpak wykorzystuje uprawnienia:

- Umożliwienie komunikacji między procesami.
- Wykorzystanie systemu wyświetlania Wayland.
- Możliwość wykorzystania systemu wyświetlania X11, jeśli istnieją problemy z Wayland.
- Możliwość wykorzystania GPU do przyspieszenia sprzętowego.
- Dostęp do systemu plików (do katalogu „Dokumenty”).
- Możliwość działania w tle.

W przypadku menedżera pakietów Snap lista uprawnień aplikacji Tabela jest następująca:

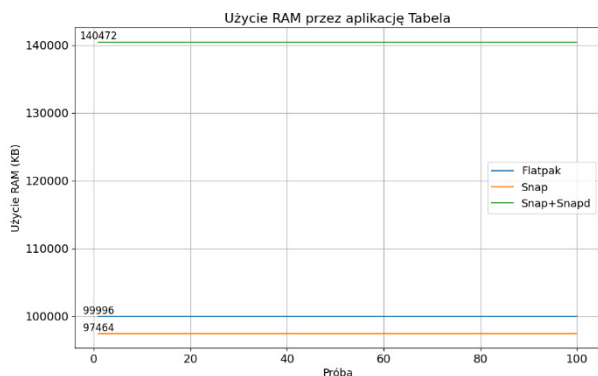
- Dostęp do systemu plików (katalog domowy użytkownika).
- Dostęp do systemu wyświetlania Wayland.
- Dostęp do systemu wyświetlania X11.
- Dostęp do zasobów środowiska graficznego.
- Dostęp do zasobów starszych środowisk graficznych.
- Dostęp do sposobu działania jako serwis sieciowy.

Pomimo dodatkowych uprawnień dostępnych z poziomu interfejsu tekstowego, to Flatpak z aplikacją Flatseal nadal umożliwia większą kontrolę nad piaskownicą w przypadku aplikacji testowej. Korzystając z aplikacji w wersji Flatpak użytkownik jest w stanie szczegółowo sprecyzować uprawnienia aplikacji dopasowując je do swoich potrzeb. Jest w stanie wybranej aplikacji wydzielić specjalny katalog, przełączyć się na serwer wyświetlania X11 (jeśli, na przykład, występują problemy z Wayland) czy wyłączyć wysyłanie wszelkich powiadomień. W przypadku menedżera pakietów Snap nie ma możliwości dostosowania większej liczby uprawnień niż wstępnie wymaga tego aplikacja. Jeśli program testowy byłby rozbudowany o dodatkowe

elementy wymagające dostępu do kolejnych modułów, na przykład kamery użytkownika lub mikrofonu, to lista uprawnień w Tabeli 1 byłaby dłuższa i konkretniejsza, tym samym nieco zmniejszając lukę pomiędzy Flatpak a Snap. Może to być szczególnie przydatne w przypadku aplikacji serwerowych, które właśnie mogą być udostępniane za pomocą menedżera pakietów Snap. Jednakże należy zaznaczyć, że dedykowane narzędzie graficzne pozwalające na kontrolę uprawnień Snap Store nie wyświetla pełnej listy uprawnień, która dopiero dostępna jest w interfejsie tekstowym. To niejako wymusza na użytkowniku zapoznanie się z niektórymi poleceniami terminalowymi, aby uzyskać jeszcze większą kontrolę nad piaskownicą.

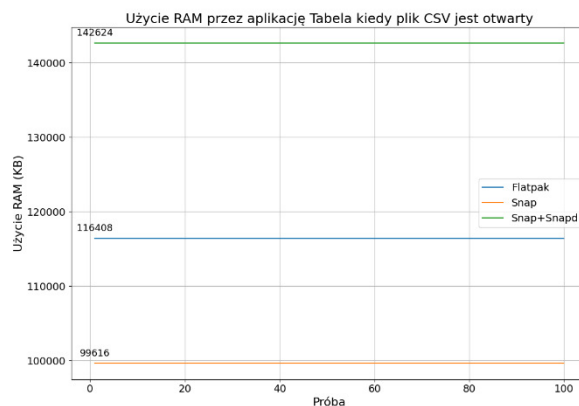
## 4.2. Wyniki dla scenariusza S2

W tym podrozdziale zostaną przedstawione wyniki dla scenariusza S2 polegającym na zbadaniu użycia pamięci RAM przez aplikację Tabela. Badanie podzielono na dwa podscenariusze. W pierwszym została uruchomiona sama aplikacja, natomiast w drugim podscenariuszu otworzono przykładowy plik CSV. Badanie wykonano za pomocą narzędzia Sysstat. Pomiaru były wykonane 100 razy w odstępach 3 sekundowych pomiędzy każdym z pomiarów. Dla wszystkich wykonanych pomiarów, wynik pozostawał ten sam, dlatego wartość pierwszego badania jest zarazem wynikiem średniej statystycznej.



Rysunek 1: Wykres przedstawiający porównanie użycia pamięci RAM przez aplikację Tabela.

Na Rysunku 1 można dostrzec, że najlepszym wynikiem, to znaczy najmniejszym użyciem pamięci RAM, cechuje się aplikacja Tabela zainstalowana za pomocą menedżera pakietów Snap. Wykorzystanie pamięci w tym przypadku wyniosło 97464 KB. Aplikacja uruchomiona za pomocą Flatpak wykorzystuje nieco więcej pamięci operacyjnej, tj. 99996 KB. Należy zaznaczyć, że aplikacje Snap do działania potrzebują uruchomionego demona Snapd. Podczas badań zajętość pamięci RAM przez tego demona jest stała i wynosi 43008 KB. Łącząc wynik demona Snapd oraz aplikacji Tabela w wersji Snap, osiągnany jest wynik 140472 KB.



Rysunek 2: Wykres przedstawiający porównanie użycia pamięci RAM przez aplikację Tabela z włączonym przykładowym plikiem CSV.

Rysunek 2 przedstawia porównanie użycia pamięci RAM przez aplikację Tabela z otworzonym przykładowym plikiem CSV. Podobnie jak w przypadku poprzedniego podscenariusza, tutaj ponownie najlepszym wynikiem wyróżnia się Snap. Co ważne, różnica pomiędzy Snap a Flatpak jest jeszcze większa. Aplikacja Tabela w wersji Snap wykorzystuje 99616 KB, kiedy to w wersji Flatpak 116408 KB. Jednakże aplikacja Snap do działania wymaga demona Snapd. Dodając użycie pamięci RAM przez demona wynik sumaryczny wynosi 142624 KB.

Aplikacja Tabela w wersji Snap cechuje się mniejszym użyciem pamięci operacyjnej RAM niż w wersji Flatpak i dotyczy to zarówno programu chwilę po uruchomieniu, jak i z otworzonym plikiem CSV. Jednakże aplikacje Snap do działania wymagają demona Snapd, który także wykorzystuje pamięć operacyjną. Sumaryczny wynik łączący zajętość aplikacji Snap i demona Snapd jest wyższy od wyniku Flatpak. Należy jednak zaznaczyć, że Snapd jest jednym demonem, który może zarządzać wieloma aplikacjami Snap. Dlatego też niekoniecznie należy go łączyć z zajętością pamięci przez samą aplikację.

## 4.3. Wyniki dla scenariusza S3

W badaniu porównano uruchamianie aplikacji Tabela zainstalowanej za pomocą Flatpak i Snap. Pierwsze uruchomienie odbyło się bezpośrednio po uruchomieniu systemu operacyjnego.



Rysunek 3: Wykres przedstawiający porównanie czasu uruchamiania aplikacji Tabela.



Wyniki badania ukazano na Rysunku 3. Uzyskane czasy uruchamiania są porównywalne i nie można zarejestrować większej różnicy. Wyjątkiem jest tutaj pierwsze uruchomienie aplikacji, gdzie w menedżerze pakietów Flatpak jest ono podobne jak w przypadku kolejnych prób, jednak w menedżerze Snap jest zauważalnie dłuższe.

Tabela 2: Dane statystyczne dotyczące czasów uruchamiania aplikacji

-	Flatpak	Snap
Średni czas uruchamiania [s]	1,0231	1,0304
Najkrótszy uzyskany czas uruchamiania [s]	1,0169	1,0165
Najdłuższy uzyskany czas uruchamiania [s]	1,0300	2,0266
Odchylenie standardowe [s]	0,0050	0,1007
Mediana [s]	1,0189	1,0177

Tabela 2 przedstawia podstawowe dane statystyczne dotyczące czasów uruchamiania aplikacji Tabela. Wartości średniej są do siebie zbliżone, choć niższą wartością cechuje się Flatpak. Ze względu na dużą liczbę prób uruchomienia, długi czas pierwszego włączenia aplikacji w wersji Snap nie wpłynął wyjątkowo zauważalnie na wynik średniej wartości. Jednakże odchylenie standardowe, a także najdłuższy czas uruchamiania aplikacji jasno wskazują na to, że wersja Snap aplikacji Tabela zapewnia mniej jednolite wyniki. Biorąc pod uwagę to, że podczas typowego korzystania z systemu operacyjnego, użytkownik uruchamia daną aplikację najczęściej raz lub kilka razy w trakcie trwania jednej sesji użytkownika, to zauważalnie dłuższy czas uruchamiania aplikacji pierwszy raz po włączeniu systemu w wersji Snap może być bardziej odczuwalny. Tym samym to właśnie wersja Flatpak zapewnia znacznie spójniejsze doświadczenia użytkownika.

Wynik badań udowadnia jedną z często wymienianych wad menedżera pakietów Snap. Pierwsze uruchomienie aplikacji po starcie systemu operacyjnego zawsze zajmuje więcej czasu. Jest to spowodowane wieloma elementami, choć najczęściej wymienia się potrzebę żądania pamięci podręcznej czcionek podczas uruchamiania. Problem został zaadresowany w 2019 roku i miał zostać rozwiązany w Snap 2.36.2 [13]. Niestety, badanie wykazuje, że problem dalej występuje.

## 5. Podsumowanie

Analizując wyniki uzyskane podczas badań można dostrzec, że pod względem wydajności Flatpak i Snap są podobnymi rozwiązaniami. W badaniu użycia pamięci RAM Snap uzyskiwał nieco lepsze wyniki, jednak są to niewielkie różnice, niemające większego wpływu na działanie komputera. W badaniu czasów uruchamiania obydwa rozwiązania uzyskiwały podobne wyniki, jednak wyjątkiem jest tutaj czas uruchomienia aplikacji Snap pierwszy raz. Jest to proces trwający zauważalnie dłużej, co w przypadku korzystania z komputera przez

typowego użytkownika, który uruchamia dany program raz, maksymalnie kilka razy podczas jednej sesji, czas włączenia aplikacji ma spore znaczenie. W tym badaniu spójniejsze wyniki oferował Flatpak.

Jak wynika z badania S1, Flatpak wraz z narzędziem Flatseal oferują o wiele większe możliwości konfiguracji uprawnień. Użytkownik zyskuje sporą kontrolę nad tym, jak aplikacja działa oraz jakie ma możliwości. W przypadku menedżera pakietów Snap opcje konfiguracji są znacznie mniejsze.

Obydwa rozwiązania, tj. Flatpak i Snap, to nowoczesne podejście do dystrybucji aplikacji o otwartym kodzie w systemach Linux. Cechuje je uniwersalność, ponieważ mogą być wykorzystywane w różnych dystrybucjach i to niezależnie od wersji systemu. Wymienione menedżery pakietów zyskują na popularności i stają się domyślną formą dostarczania oprogramowania w coraz to większej liczbie dystrybucji. Wybierając aplikacje Flatpak lub Snap użytkownik zyskuje bardzo stabilną pracę, wygodę oraz nowoczesne formy dbania o bezpieczeństwo i prywatność. Użytkownik uzyskuje również sporą kontrolę nad oprogramowaniem dzięki systemowi uprawnień.

Flatpak i Snap to również spore ułatwienie dla deweloperów. Są to uniwersalne menedżery pakietów, więc przygotowując pakiet na jeden z nich (lub na oba) jednocześnie trafia się do ogromnej liczby dystrybucji, a nie tylko do jednej (lub jednej z rodzin dystrybucji). Deweloper ma również większą kontrolę nad środowiskiem wykonawczym. To powinno ograniczyć liczbę zgłaszanych błędów, które mogłyby wynikać z wykorzystywania niekompatybilnych składników systemowych, na przykład bibliotek.

## Literatura

- [1] Aplikacja Tabela w centralnym repozytorium Flathub, <https://flathub.org/pl/apps/eu.cichy1173.tabela>, [07.09.2023].
- [2] Aplikacja Tabela w centralnym repozytorium Snap Store, <https://snapcraft.io/tabela>, [07.09.2023].
- [3] Repozytorium zawierające kod źródłowy aplikacji Tabela, <https://codeberg.org/cichy1173/tabela-flatpak>, [16.09.2023].
- [4] Repozytorium zawierające plik manifestu dla sklepu z aplikacjami Flathub, <https://github.com/flathub/eu.cichy1173.tabela>, [16.09.2023].
- [5] Repozytorium zawierające plik manifestu dla sklepu z aplikacjami Snap Store, <https://codeberg.org/cichy1173/Tabela>, [16.09.2023].
- [6] D. Spinellis, Package Management Systems, IEEE Computer Society 29 (2012) 84-86, <https://doi.org/10.1109/MS.2012.38>.
- [7] P. Abate, R. Di Cosmo, R. Treinen, S. Zacchiroli, A modular package manager architecture, Information and Software Technology 55 (2012) 459-474, <https://doi.org/10.1016/j.infsof.2012.09.002>.

- [8] A. Athalye, R. Hristov, T. Nguyen, Q. Nguyen, Package Manager Security, project report, Massachusetts Institute of Technology (2014).
- [9] E. Kokot, Distribution-agnostic package management on Linux with Flatpak, thesis, University of Ljubljana (2022).
- [10] N. Fraser, Flatpak is Not the Future, <https://ludocode.com/blog/flatpak-is-not-the-future>, [07.09.2023].
- [11] H. Rana, Response to "Flatpak Is Not the Future", <https://theevilskeleton.gitlab.io/2022/05/16/response-to-flatpak-is-not-the-future.html>, [07.09.2023].
- [12] T. Dunlap, W. Enck, B. Reaves, A Study of Application Sandbox Policies in Linux, Proceedings of the 27th ACM on Symposium on Access Control Models and Technologies, (2022) 19-30.
- [13] Uruchamianie aplikacji Snap, <https://www.omgubuntu.co.uk/2019/03/the-cause-of-slow-snap-app-startup-times-has-been-identified>, [07.09.2023].