

Analiza czasowa wydajności systemów Windows 10 oraz Windows 8.1 wykonana na podstawie aplikacji mobilnej

Jacek Chmiel*, Maria Skublewska-Paszkowska

Politechnika Lubelska, Instytut Informatyki, Nadbystrzycka 36B, 20-618 Lublin, Polska

Streszczenie. W artykule przedstawiono analizę czasową wydajności systemów Windows 10 oraz Windows 8.1. Przeprowadzone badania skupiały się na porównaniu wydajności elementów systemu urządzenia mobilnego, które są ważne z punktu widzenia użytkownika a mianowicie na szybkości procesora, przepustowości RAM i pamięci masowej, szybkości generowania obrazu przez procesor graficzny oraz czasie dostępu do urządzeń. Na potrzeby analizy została stworzona aplikacja mobilna do pomiaru wyżej wymienionych elementów. Postawione w artykule hipotezy badawcze zostały zweryfikowane i częściowo udowodnione.

Słowa kluczowe: analiza czasowa; wydajność; systemy mobilne; Windows 10; Windows 8.1

*Autor do korespondencji.

E-mail: chmiel.jacek1990@gmail.com

Time analysis of the performance of Windows 10 and Windows 8.1 based on mobile application

Jacek Chmiel*, Maria Skublewska-Paszkowska

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

Abstract. This article presents time analysis of performance of Windows 10 and Windows 8.1. Presented research was mainly focused on performance comparison of mobile device system elements which are relevant from end user point of view like main processor, RAM and mass storage throughput, image generation speed by the graphics processor and access time to devices. In order to measure the performance of the above elements a mobile benchmark application was created. The hypotheses set in the article have been verified and partially confirmed.

Keywords: performance; time analysis; mobile systems; Windows 10; Windows 8.1

*Corresponding author.

E-mail address: chmiel.jacek1990@gmail.com

1. Wstęp

Urządzenia mobilne na przestrzeni ostatnich kilkunastu lat zyskały na popularności tak bardzo, że trudno sobie dzisiaj wyobrazić ich brak w codziennym życiu. Możliwości współczesnych urządzeń mobilnych są zbliżone do tych, jakie oferują komputery stacjonarne czy laptopy a przy tym ich możliwości przenoszenia są dużo większe. Wobec tej popularności i powszechnego występowania bardzo ważna jest wydajność całego systemu urządzenia mobilnego z punktu widzenia użytkownika w czasie codziennego pracy.

Celem pracy jest analiza czasowa wydajności systemów Windows 10 i Windows Phone 8.1. Aby możliwe było wykonanie analizy wydajności po tym kątem należało przeprowadzić eksperyment badawczy, który pozwoli zmierzyć czas wykonywania wybranych operacji za pomocą urządzenia mobilnego, a następnie porównać uzyskane wyniki pomiędzy systemami. Na potrzeby badań zostało stworzone autorskie, uniwersalne narzędzie badawcze, które za pomocą jednolitego zestawu testów jest w stanie przetestować obydwie systemy.

2. Badania literaturowe

Wydajność może być definiowana na wiele sposobów w zależności od kontekstu, w jakim chce się ją rozpatrywać [1]. Według Arnolda O. Allena wydajność jest miarą tego jak dobrze komputer wykonuje pracę, którą powinien wykonywać rozumianą jako wykonywanie poleceń [2]. W niniejszym artykule wydajność systemu operacyjnego jest rozumiana jako czas wykonywania operacji, które bezpośrednio wpływają na jego odbiór oraz komfort pracy z nim przez użytkownika końcowego. Istnieje szereg sposobów pomiaru wydajności systemów komputerowych jednak na początek należy proces analizy wydajności poprzedzić zebraniem wiedzy na temat badanego systemu jak i zdefiniować elementy, jakie należy poddać analizie oraz wyznaczyć kryteria ich oceny [3].

System operacyjny jest programem komputerowym, który pośredniczy między użytkownikiem komputera a jego podzespołami. Jego głównym celem jest stworzenie środowiska do uruchamiania programów, ich kontroli oraz dzielenia dostępnych zasobów. W tym celu system zajmuje się planowaniem i przydzielaniem czasu procesora, kontrolą

i przydziałem pamięci operacyjnej oraz obsługą sprzętu komputerowego [4].

Po przeprowadzeniu studiów literaturowych można zauważyć, że występuje niewiele prac w literaturze naukowej bezpośrednio porównujących jakiegokolwiek dwa systemy operacyjne, czy to mobilne czy innego rodzaju. W przeważającej części są to analizy badające wydajność pojedynczych programów, algorytmów lub protokołów działających na urządzeniach mobilnych bądź możliwości jakie dają te urządzenia w postaci wsparcia w zbieraniu danych np.: do celów medycznych lub wspierania procesu nauki [5-8]. Prowadzone są także badania analizujące wydajność aplikacji mobilnych zbudowanych za pomocą rozwiązań międzyplatformowych [9].

Jednymi z nielicznych badań porównującymi mobilne systemy operacyjne pod kątem wydajności są badania przeprowadzone przez badaczy z uniwersytetu Maharashtra, opisujące porównanie mobilnego systemu operacyjnego Ubuntu Toch i Android [10] lub badania wydajności wbudowanych systemów mobilnych [11]. Nie zostały jak do tej pory przeprowadzone żadne badania wydajności systemów mobilnych firmy Microsoft. W związku z tym jest w pełni uzasadnione podjęcie tematu czasowej analizy wydajności systemu Windows 10 na urządzeniach mobilnych.

Obszarem objętym badaniem będzie wydajność systemów Windows 10 i Windows Phone 8.1 na podstawie aplikacji mobilnej. Do eksperymentu badawczego zostały wybrane dwa modele urządzeń mobilnych Nokia Lumia 830 i Nokia Lumia 930. Na obydwu urządzeniach były zainstalowane systemy Windows 10 i Windows Phone 8.1, tak więc do testów posłużyły 4 różne fizyczne urządzenia.

Zostały postawione następujące hipotezy badawcze. Wydajność systemu operacyjnego Windows Phone 8.1 będzie większa od wydajności systemu operacyjnego Windows 10 Mobile na urządzeniu mobilnym Nokia Lumia 830. Wydajność systemu operacyjnego Windows Phone 8.1 będzie większa od wydajności systemu operacyjnego Windows 10 Mobile na urządzeniu mobilnym Nokia Lumia 930.

3. Metoda badań

3.1. Obiekt badań

Obiektem badań są mobilne systemy operacyjne Windows 10 i Windows Phone 8.1. Na potrzeby analizy ich wydajności zostały wybrane elementy systemu urządzenia mobilnego, których szybkość i efektywność pracy ma wpływ na jego użytkowanie. Są to procesor, pamięć RAM, pamięć masowa, procesor graficzny oraz czas dostępu do interfejsu API, służący do obsługi funkcjonalności typowych dla urządzeń mobilnych.

Systemy Windows 10 i Windows Phone 8.1 zostały przetestowane na urządzeniach mobilnych o następującej specyfikacji:

- Nokia Lumia 830 posiada ekran o przekątnej 5 cali, pamięć operacyjną o rozmiarze 1 gigabajta i czterordzeniowy procesor Qualcomm Snapdragon 400 o taktowaniu pojedynczego rdzenia 1.2 gigaherca.
- Nokia Lumia 930 posiada 5 calowy ekran jednak ma 2 gigabajty pamięci operacyjnej i czterordzeniowy procesor Qualcomm Snapdragon 800 o taktowaniu 2.2 gigaherca.

3.2. Opis narzędzia badawczego

Na potrzeby przeprowadzenia badań zostało stworzone uniwersalne narzędzie badawcze, które jednym zestawem testów symulujących obciążenie systemu jest w stanie przetestować wydajność systemów Windows 10 oraz Windows Phone 8.1. Bada ono w następujący sposób elementy wybrane systemu urządzenia mobilnego.

W celu przetestowania procesora zostały utworzone algorytmy testujące instrukcje arytmetyczne operujące na liczbach stałoprzecinkowych i zmiennoprzecinkowych. Najprostszym sposobem na obciążenie procesora w stu procentach jest uruchomienie kolejnych operacji w pętli a następnie pomiar czasu ich wykonania. Na potrzeby tego badania została określona liczba obrotów pętli na sto milionów a w każdym z nich są wykonywane operacje dodawania, odejmowania, mnożenia i dzielenia.

W celu przetestowania przepustowości zapisu i odczytu pamięci RAM został stworzony algorytm, który zapisuje do pamięci i z niej wczytuje wcześniej przygotowany plik z danymi o rozmiarze 16 MB, który powstaje w czasie instalacji aplikacji testującej. Plik ten jest w teście zapisu iteracyjnie zapisywany do pamięci, a w teście odczytu zapisywany jeden raz, a następnie określoną ilość razy wczytywany z pamięci. Znając ilość iteracji oraz rozmiar pliku obliczana jest przepustowość pamięci podana w MB/s.

Do testów przepustowości pamięci masowej również jest wykorzystywany wcześniej przygotowany plik o rozmiarze 16 MB. W przypadku testu odczytu z pamięci masowej jest on odczytywany określoną ilość razy z pamięci masowej telefonu. Znając ilość iteracji obliczana jest przepustowość odczytu z pamięci masowej w MB/s. W przypadku testu zapisu plik ten jest wczytywany do pamięci RAM określoną liczbę razy, dynamicznie ustalaną w czasie wykonywania programu, a następnie tak przygotowana paczka danych jest zapisywana do pamięci masowej. Tutaj również przepustowość jest podawana w MB/s.

W celu przetestowania procesora graficznego została użyta biblioteka Win2D wspierająca rysowanie grafiki [12]. Wydajność jest tutaj badana jako średni czas generowania obrazu wyświetlanego na ekranie urządzenia mobilnego. Test trwa 30 sekund i co jedną sekundę jest dodawany do rysowanej sceny jedna losowa elipsa, co zwiększa obciążenie

procesora graficznego. Wynikiem testu jest średnia liczba klatek na sekundę.

Funkcjonalności specyficzne dla urządzeń mobilnych są testowane algorytmem mierzącym czas dostępu do metod API, służących do obsługi akcelerometru, notyfikacji użytkownika i wibracji. Test ten wskazuje na szybkość obsługi urządzeń na danym systemie.

Sposób wykonania narzędzia badawczego pozwala na uruchomienie wszystkich wyżej wymienione testów w kolejności sekwencyjnej za pomocą naciśnięcia jednego przycisku na głównym ekranie aplikacji. Po przeprowadzeniu wszystkich testów poszczególne wyniki osiągnięte w każdym z nich są prezentowane na interfejsie użytkownika.

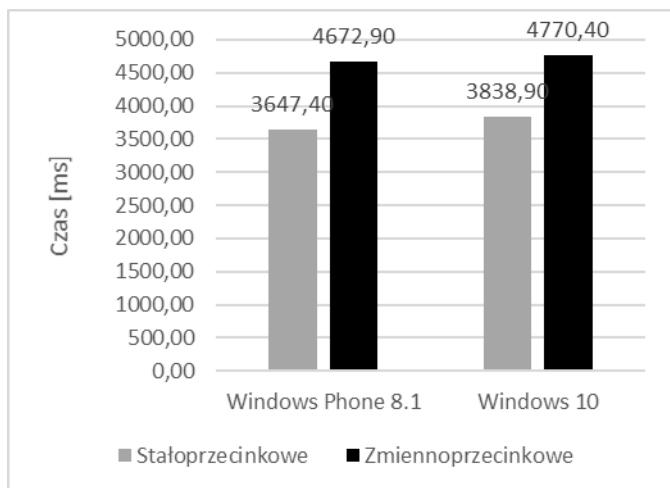
3.3. Procedura eksperymentów

Przeprowadzono po 10 testów wydajności na każdym urządzeniu co dało łącznie 40 testów. Sposób wykonania aplikacji powoduje, że aplikacja po naciśnięciu przycisku start na interfejsie użytkownika wykonuje wszystkie zaprojektowane testy sekwencyjnie.

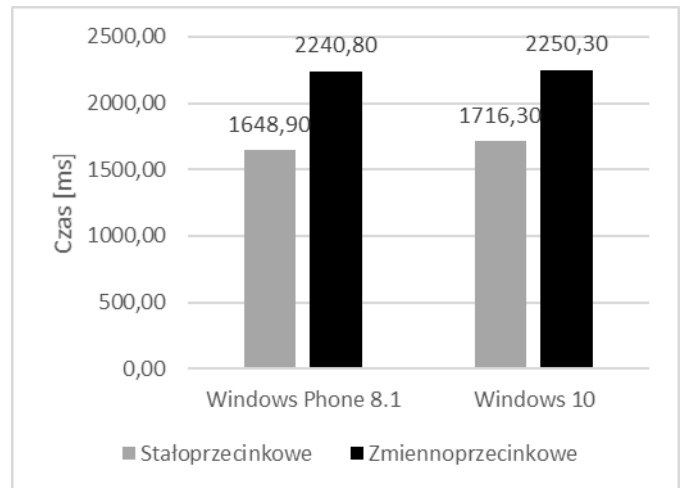
Każdy test był wykonywany po świeżym uruchomieniu aplikacji testującej, w celu wyeliminowania optymalizacji, jakie środowisko uruchomieniowe stosuje dla aplikacji na platformie .NET. Miało to na celu w jak największym stopniu przetestować różnicę pomiędzy dwoma systemami. Po każdym wykonanym teście spisywane były rezultaty uzyskane za jego pomocą.

4. Wyniki badań

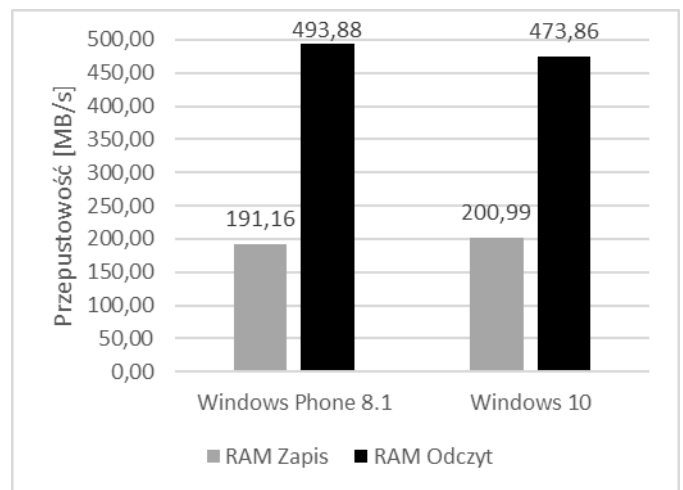
Na rysunkach 1-11 zostały przedstawione uśrednione wyniki ze wszystkich przeprowadzonych testów wydajności.



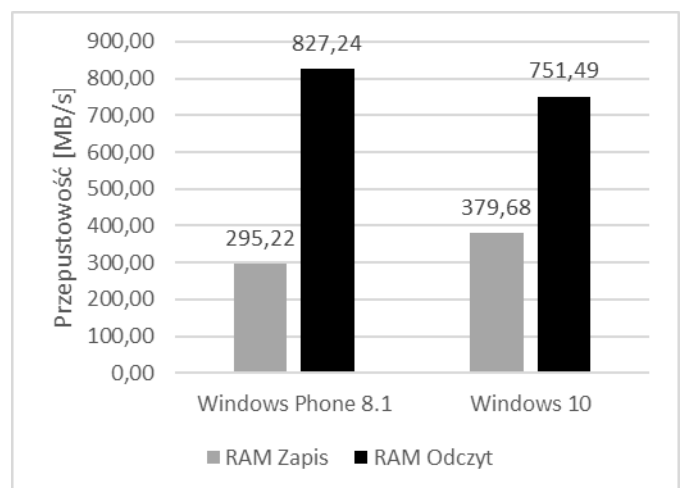
Rys. 1. Średnie czasy testów procesora na urządzeniu Nokia Lumia 830



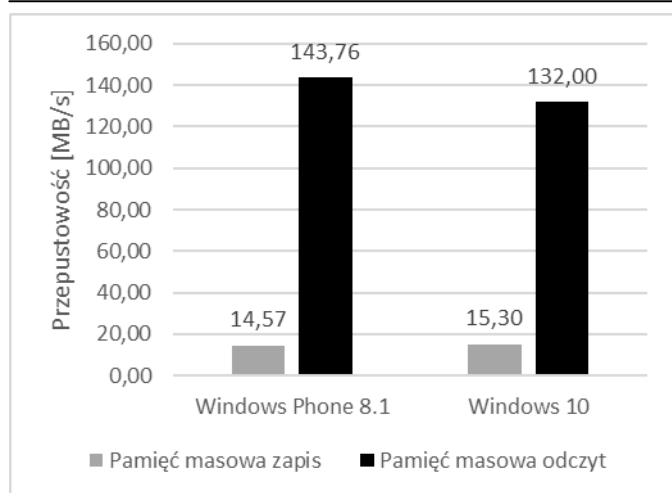
Rys. 2. Średnie czasy testów procesora na urządzeniu Nokia Lumia 930



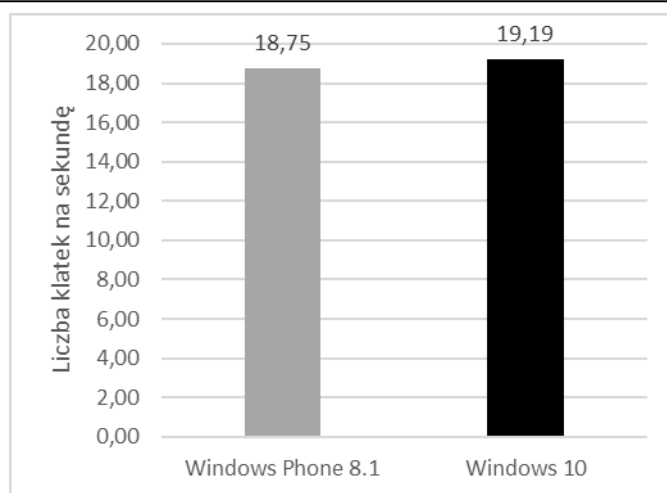
Rys. 3. Średnia przepustowość pamięci RAM na urządzeniu Nokia Lumia 830



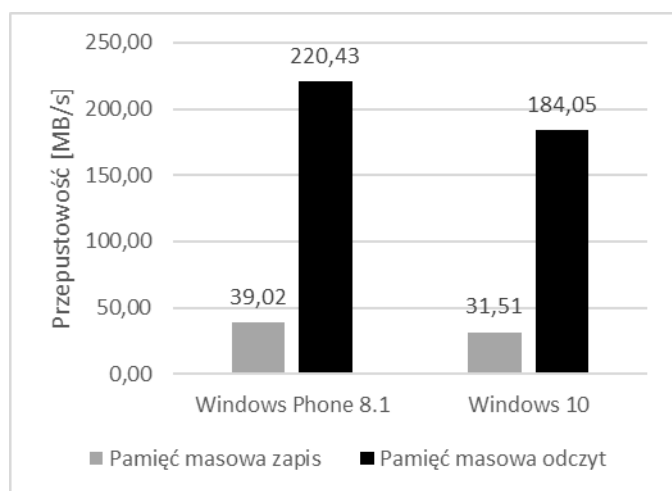
Rys. 4. Średnia przepustowość pamięci RAM na urządzeniu Nokia Lumia 930



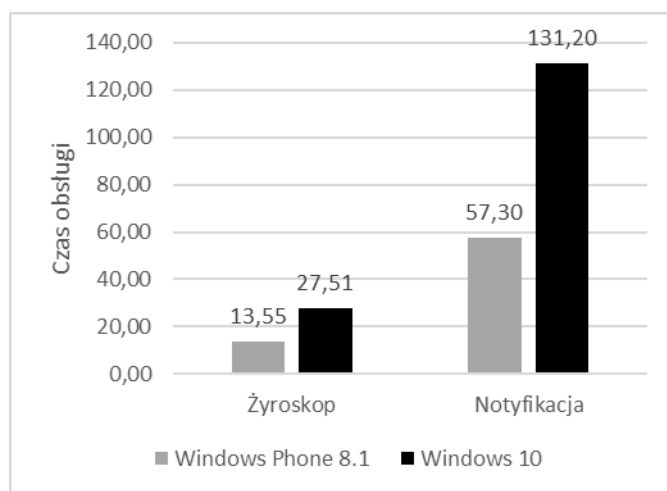
Rys. 5. Średnia przepustowość pamięci masowej na urządzeniu Nokia Lumia 830



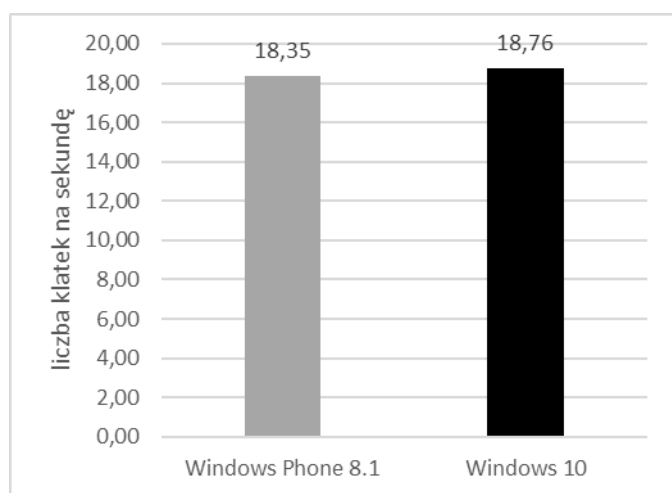
Rys. 8. Średnia liczba klatek na sekundę osiągnięta w teście procesora graficznego na urządzeniu Nokia Lumia 930



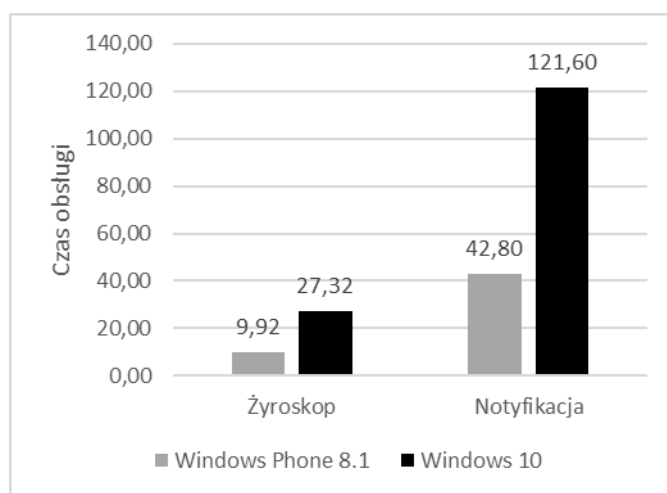
Rys. 6. Średnia przepustowość pamięci masowej na urządzeniu Nokia Lumia 930



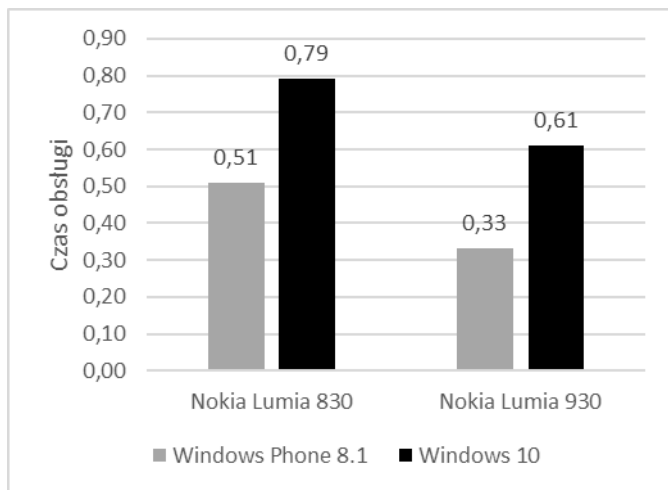
Rys. 9. Średnie czasy obsługi API żyroskopu i notyfikacji na urządzeniu Nokia Lumia 830



Rys. 7. Średnia liczba klatek na sekundę osiągnięta w teście procesora graficznego na urządzeniu Nokia Lumia 830



Rys. 10. Średnie czasy obsługi API żyroskopu i notyfikacji na urządzeniu Nokia Lumia 930



Rys. 11. Zestawione rednie czasy obsługi API wibracji obydwu systemach i urządzeniach

Na rysunkach 1 i 2 zostały uzyskane wyniki badania procesora jako czas wykonywania operacji arytmetycznych z użyciem liczb stałoprzecinkowych i zmiennoprzecinkowych. W teście tym na obydwu urządzeniach lepszy rezultat od Windows 10 Mobile uzyskał system Windows Phone 8.1. Na urządzeniu Nokia Lumia 830 był o 4,99% szybszy w teście na liczbach stałoprzecinkowych i o 2,04% szybszy w teście na liczbach zmiennoprzecinkowych. Na urządzeniu Nokia Lumia 930 było szybszy o odpowiednio 3,93% oraz 0,42%.

Na rysunkach 3 i 4 pokazane są wyniki przepustowości pamięci RAM zarówno odczytu z pamięci jak i zapisu do niej. Windows Phone 8.1 osiągnął lepszą przepustowość odczytu na obydwu urządzeniach o 4,49% w przypadku urządzenia Nokia Lumia 830 i o 9,26% w przypadku Nokia Lumia 930. Jednak Windows 10 osiągnął lepsze rezultaty w teście zapisu do pamięci o 5,14% na urządzeniu Nokia Lumia 830 i o 28,61% na urządzeniu Nokia Lumia 930.

Rysunki 5 i 6 pokazują wyniki testów przepustowości pamięci masowej. Na urządzeniu Nokia Lumia 830 Windows 10 w teście zapisu do pamięci osiągnął o 5,14% lepszy wynik od Windows Phone 8.1. W pozostałych przypadkach ten drugi osiągnął lepsze rezultaty. Na urządzeniu Nokia Lumia 930 zapis do pamięci masowej było o 19,26% szybszy, natomiast odczyt o 9,16% szybszy. Na urządzeniu Nokia Lumia 830 osiągnął odczyt o 4,05% szybszy.

Na rysunkach 7 i 8 pokazane są wyniki testu procesora graficznego przedstawione jako średnia ilość klatek na sekundę. Na obydwu urządzeniach lepsze rezultaty osiągnął system Windows 10 Mobile. Dla Nokia Lumia 830 rezultat był o 2,24% a dla Nokia Lumia 930 2,36%.

Rysunki 9-11 pokazują wyniki testów czasów dostępu do API systemów urządzenia mobilnego żyroskopu, wibracji i notyfikacji. We wszystkich tych testach Windows 10 osiągnął dużo słabsze wyniki. Na urządzeniu Nokia Lumia 830 były one słabsze odpowiednio o 51,33%,

35,44% i 66,33%. Na urządzeniu Nokia Lumia 930 były one słabsze odpowiednio o 63,69%, 45,90% oraz o 64,8%.

5. Dyskusja wyników

Po przeprowadzonej analizie uzyskanych wyników postawione hipotezy zostały częściowo potwierdzone. System Windows 10 okazał się niemalże we wszystkich testach mniej wydajny od systemu Windows Phone 8.1. Największe różnice pomiędzy obydwoma systemami zostały ukazane w testach dostępu do urządzeń (Rys. 9-11). W tych testach Windows Phone 8.1 osiąga czasy około dwa niższe na urządzeniu Nokia Lumia 830 oraz około trzy razy niższe na urządzeniu Nokia Lumia 930. Windows 10 osiągnął lepsze rezultaty na obydwu urządzeniach jedynie w teście wydajności procesora graficznego (Rys. 7-8) oraz w teście zapisu do pamięci RAM (Rys. 3-4). Na urządzeniu Nokia Lumia 830 osiągnął dodatkowo lepszy wynik w teście zapisu do pamięci masowej (Rys. 5). Wyniki uzyskane podczas przeprowadzania analizy porównawczej potwierdzają przypuszczenia autora o niewielkiej wariancji pojedynczych rezultatów, która w przypadku niektórych testów wynosi około 5% średnich wartości uzyskanych w tych testach. Jest to podyktowane izolacją procesu, który wykonuje kod aplikacji na systemach mobilnych oraz przydzielania mu całych dostępnych zasobów sprzętowych urządzenia mobilnego przez system. Wyjątkiem tutaj jest limit przydzielanej pamięci operacyjnej dla pojedynczej aplikacji, różny dla każdego systemu oraz dostępnej pamięci na urządzeniu.

Otrzymana różnica w wydajności wynikać może z innego zestawu sterowników urządzeń, implementacji jądra obydwu systemów jak i samej praformy .NET oraz jej komponentów, takich jak zbieracz śmieci lub kompilator JIT (ang. Just in Time). Wszystkie te czynniki mogą powodować większy narzut podczas obsługi poszczególnych elementów systemu urządzenia mobilnego. Jednoznaczne stwierdzenie, który z powyższych elementów miał największy wpływ na ukazaną różnicę wydajności jest niemożliwe ze względu na zamknięcie ich źródeł przez firmę Microsoft.

6. Wnioski

Rezultaty przeprowadzonych badań dają obraz różnic pomiędzy obydwoma systemami i mogą posłużyć jako wskazówka dla osób chcących nabyć urządzenie mobilne z jednym z systemów mobilnych firmy Microsoft lub mogą zostać wykorzystane przez samych twórców systemu, używając przygotowane narzędzie do porównania wydajności. Do przedstawionej analizy porównawczej zostały wykorzystane dwa identyczne urządzenia mobilne, w związku z tym, ukazane różnice w wydajności mogą wynikać z odmiennej wewnętrznej budowy jądra systemów oraz platformy .NET jak również z innego zestawu sterowników.

W przyszłości pracując nad następną analizą można było dokonać pomiaru całkowitego czasu wykonania wszystkich testów jako kolejny element badań. Kolejną sugestią jest również głębsze poznanie API

dostępnego do obsługi grafiki 2D. Aktualnie dostępny interfejs programisty nie daje możliwości próbkowania szybkości wyświetlania grafiki na ekranie urządzenia mobilnego co wymusza pomiar czasu wykonywania kodu z poziomu aplikacji, zamiast pobierania czasu generowania grafiki z poziomu API.

Literatura

- [1] Lilja D.J.: *Measuring Computer Performance: A Practitioner's guide*, Cambridge University Press, 2005.
- [2] Allen A.O.: *Computer Performance Analysis with Mathematica*. Academic Press, 1994.
- [3] Jain R.K.: *Art of Computer Systems Performance Analysis Techniques For Experimental Design Measurements Simulation And Modeling*. Wiley, 1991.
- [4] Hoare C.A.R.: *Monitors: An Operating System Structuring Concept*. *Communication of the ACM*, 10 (1974), 549-557.
- [5] Sung Y., Chang K., Liu T.: The effects of integrating mobile device with teaching and learning on students' learning performance: a meta-analysis and research synthesis, *Computer & Education*, 3 (2016), 252–275.
- [6] Shahabi S., Ghazvini M., Bakhtiarian M.: A modified algorithm to improve security and performance of AODV protocol against black hole attack. *Springer Wireless Network*, 5 (2015), 1505-1511.
- [7] V. D. Blondel, A. Decuyper, G. Krings.: A survey of results on mobile phone datasets analysis, *EPJ Data Science*, 1 (2015), 1–55.
- [8] Mazomenos E.B., Biswas D., Acharyya A., Chen T., Maharatna K., Rosengarten J., Morgan J., Curzen N.: A Low-Complexity ECG Feature Extraction Algorithm for Mobile Healthcare Applications, *IEEE Journal of Biomedical and Health Informatics*, 2 (2013), 459-469.
- [9] Latif M., Lakhri Y., Nfaoui E.H., Es-Sbai, N.: Cross platform approach for mobile application development: A survey. *IT4OD* (2016), <http://dx.doi.org/10.1109/IT4OD.2016.7479278>.
- [10] Shaikh A.S, Inamdar M.U.: Performance Analysis: Ubuntu Touch v/s Android OS. *International Journal of Science, Engineering and Technology Research*, 3 (2014), 1795-1800.
- [11] Park S., Park Y., Choi J.: Performance Analysis of Embedded OS in ARM Platform, *ICTC 2012*
- [12] <https://blogs.windows.com/buildingapps/2014/09/05/introducing-win2d-gpu-accelerated-2d-graphics-programming-in-the-windows-runtime/#721RzUqREPKGhSRb.97>, [27.02.2018]