

## 1. Opis

*Terminal Clock* umożliwia użytkownikowi terminala systemu Linux sprawdzenie aktualnej godziny poprzez zegar analogowy posiadający trzy wskazówki. Wskazówki krótka, średnia oraz długa odpowiadają oznaczeniu odpowiednio aktualnej godziny, minuty oraz sekundy. Zegar rysowany jest przy użyciu znaków kodu ASCII stosując rozwiązanie płaszczyzny kartezjańskiej opisaną w dalszej części dokumentu.

## 2. Geneza

Świadomość istnienia polecenia `date` skłoniła mnie do rozważania nad tym, jakie są ograniczenia tekstowego trybu pracy. Pomyślałem wtedy właśnie o takim zegarze wskazówkowym, który zaprogramowany dla środowiska konsolowego byłby czymś w rodzaju gry – może i nic nie wnosi do produktywności, ale jest swego rodzaju miodek dla oka; istnym pokazem ludzkiego sprytu.

## 3. Działanie

Aby móc rysować po konsoli, należy najpierw mieć do tego celu odpowiednie struktury. Postanowiłem nie opierać się o żadne zewnętrzne, tylko stworzyć własną miniaturę i jednocześnie wbudowaną w projekt bibliotekę pozwalającą na traktowanie wycinka bufora konsoli jako płaszczyzny kartezjańskiej.

Pojedynczy punkt znajdujący się na owej płaszczyźnie zajmuje dwa znaki poziomo i jeden znak pionowo; powodem takiego rozwiązania jest większe podobieństwo dwóch znaków zapisanych koło siebie do kształtu kwadratu w większości konfiguracji konsolowych. Czasami nawet zdarza się tak, że jeden znak zachowuje swoją szerokość i wysokość w proporcji 1:2, co oznacza że dwa znaki zapisane koło siebie zajmują obszar kwadratowy. Dlaczego zależy nam na kształcie kwadrato-podobnym? Otóż punkty płaszczyzny wykazujące taki kształt, po narysowaniu zachowują proporcję szerokości do wysokości wycinka bufora. Przykładowo definiując płaszczyznę 10x10, po narysowaniu oczekujemy widoku kwadratu – w większości konsoli jest to jednak nieosiągalne i zawsze widoczne będzie niewielkie spaczenie proporcji wycinka bufora (dające w rezultacie prostokątny obszar).

Punkty płaszczyzny można swobodnie odczytywać i modyfikować. Samą płaszczyznę, jak już wspomniano w poprzednim akapicie, można narysować. Punkty są zapisywane w pamięci takim sposobem, że funkcja odpowiedzialna za rysowanie ma za zadanie jedynie traktować bufor punktów jako ciąg znaków, który następnie jest przekierowywany do standardowego strumienia wyjściowego, którym domyślnie jest konsola.

Posiadając funkcjonalność płaszczyzny, jesteśmy już w stanie realizować niemal każdą prezentację graficzną, a w kontekście tego projektu – wizualizację analogowego zegara. Najpierw rysowana jest tarcza, potem liczby odpowiadające godzinom 1-12 i na koniec trzy wskazówki: krótka (godzina), średnia (minuta) oraz długa (sekunda). Oczywiście wszystkie wskazówki są zgodne z aktualną godziną wnioskowaną przy użyciu czasu unixowego dostępnego za pośrednictwem biblioteki `chrono`. Ruch wskazówek jest ciągły – nie polega on na nagłych przeskokach.

#### 4. Użycie

Narzędzie *Terminal Clock* działa jak polecenie. Po zbudowaniu projektu narzędziem *make*, w katalogu *bin* zostanie osadzony linuksowy plik wykonywalny pod nazwą *clock*. To właśnie przy jego użyciu możemy wyświetlać zegar, opcjonalnie kontrolując jego aspekty jakimi są odświeżanie (flaga *update*, automatyczne przerysowanie co sekundę) oraz średnica (parametr *diameter*, średnica tarczy w znakach). Poniżej przedstawiona została składnia polecenia *clock*, wykorzystując standardowy zapis tekstu pomocniczego polecenia dla systemu Linux:

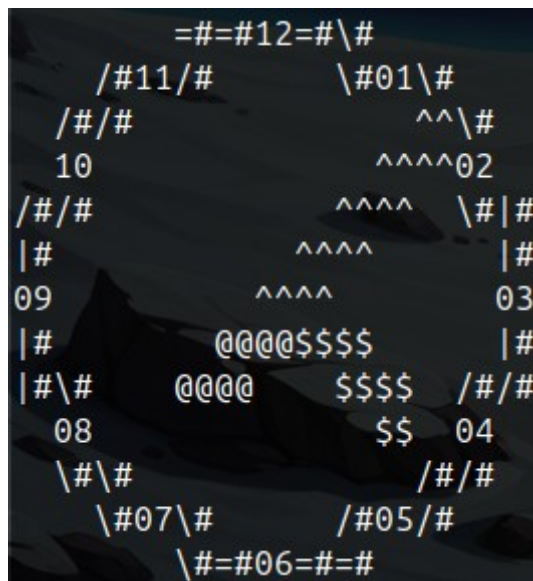
1. `clock --help`
2. `clock [--update] [diameter]`

Dla przypomnienia: parametry zakleszczone w kwadratowe nawiasy nie są wymagane, ale jak już chcemy je użyć, to muszą zachować nadaną kolejność.

#### 5. Przykłady

Wstrzymuję się od przykładów z flagą *update*, gdyż i tak nie można by zaobserwować efektu odświeżania.

##### 5.1. Wywołanie `clock 13`



5.2. Wywołanie `clock 25` równoważne niepodaniu parametru średnicy

