PODSUMOWANIE PIERWSZEGO SPRINTU

Wybór języka oprogramowania i jego środowiska :

Po przeprowadzeniu ankiety i wspólnej naradzie, wybraliśmy C++. Uważamy, że ten język będzie najbardziej odpowiedni dla naszego projektu, również największą wiedzę posiadamy w tym języku.

Środowisko, które wybraliśmy nie jest zbytnio zaawansowane, zwykły notatnik w Linuksie. Kompilowanie, linkowanie na początku naszej pracy będziemy wykonywać za pomocą komend w terminalu linuksa.

Zgłębiliśmy dostatecznie naszą wiedzę na temat kodeków i ogólnie konwersji muzyki. W wiki na gicie są zamieszczone informacje, do których każdy będzie miał dostęp w każdej chwili.

Zaczynamy poszukiwanie odpowiedniej biblioteki, która znacznie ułatwi nam prace nad projektem. Pisanie własnych funkcji w temacie konwersji audio jest znacznie bardziej zaawansowaną sprawą, której nie jesteśmy w stanie wykonać. Korzystanie z bibliotek dostępnych w internecie będzie naszym atutem w złożoności kodu ale także czasu przeznaczonego na jego pisanie. Ograniczymy się tym samym do rozumienia funkcji, przez pryzmat użytkowania a nie pisania od zera.

Kwestia wybrania odpowiedniej biblioteki do tego projektu, nie była tak oczywista do rozstrzygnięcia.

Głowna strona naszej biblioteki : <http://lame.sourceforge.net/about.php>

Będziemy z niej głownie korzystać.

*“LAME is a high quality MPEG Audio Layer III (MP3) encoder licensed under the LGPL.”*

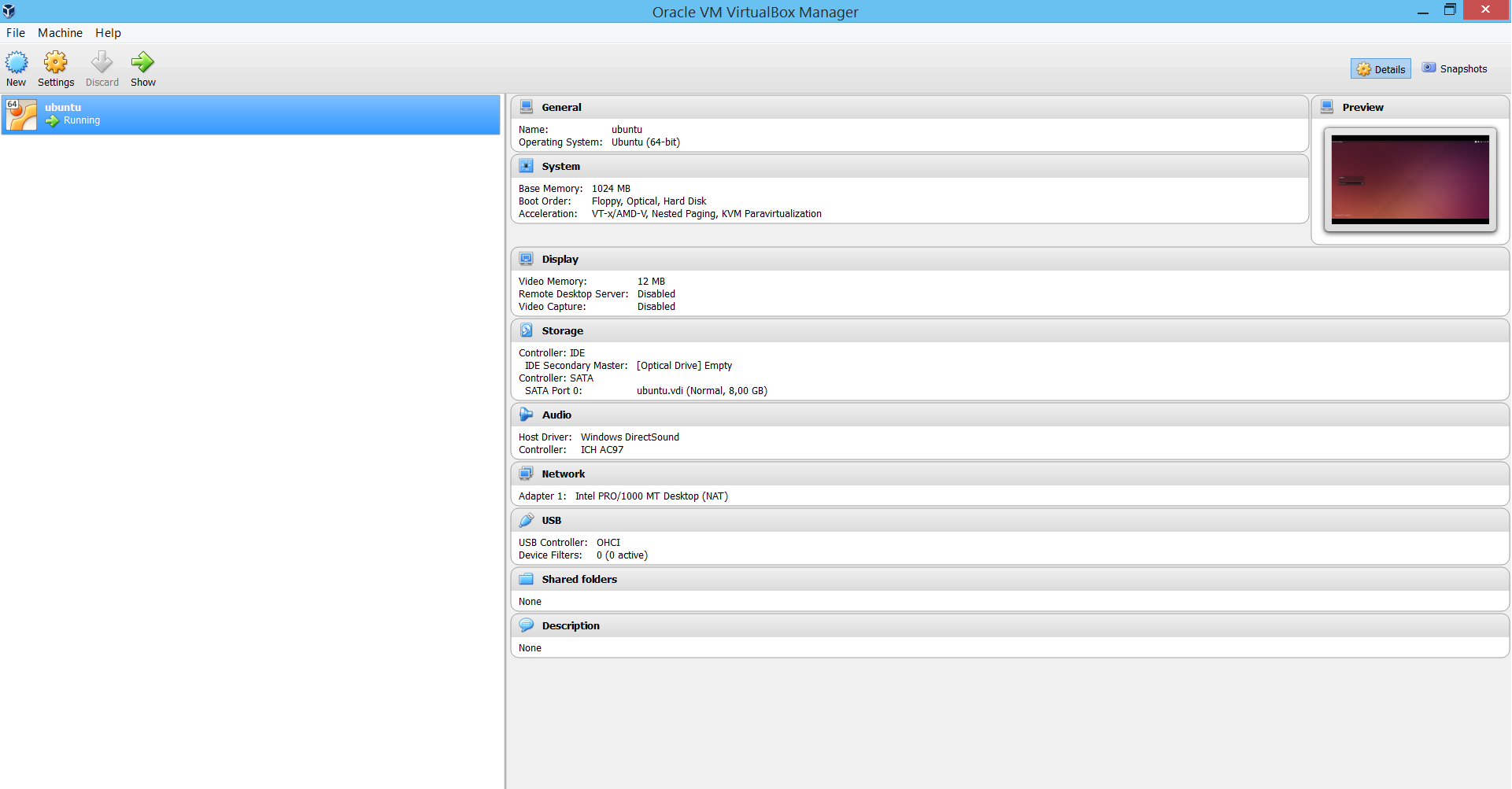
Odskakując trochę, dla ludzi nie posiadających linuksa :

Można go zainstalować na wirtualnej maszynie, my korzystamy z VirtualBoxa.

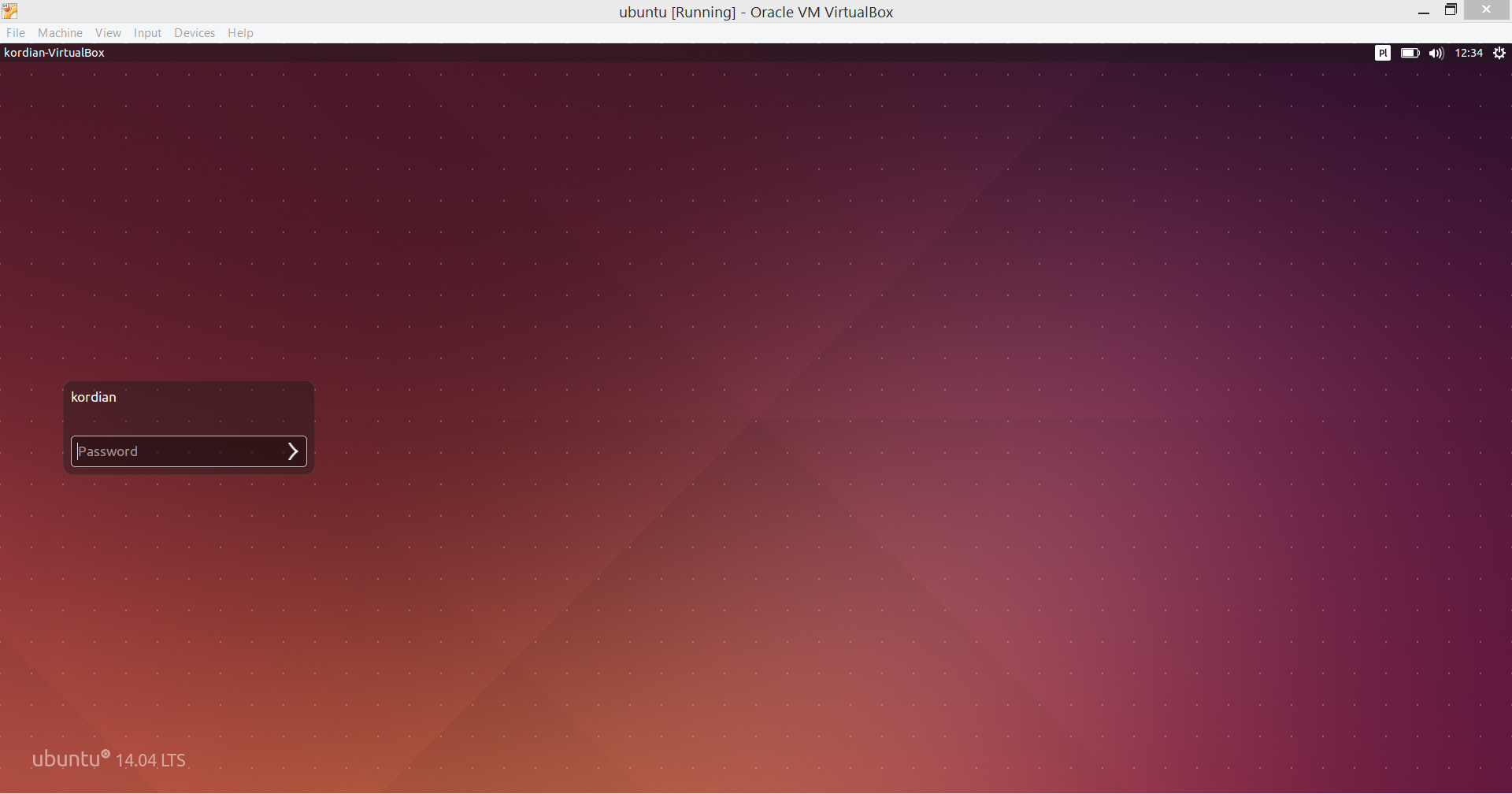
Link do pobrania : <https://www.virtualbox.org/>

Pobieramy też Ubuntu, <http://www.ubuntu.com/>

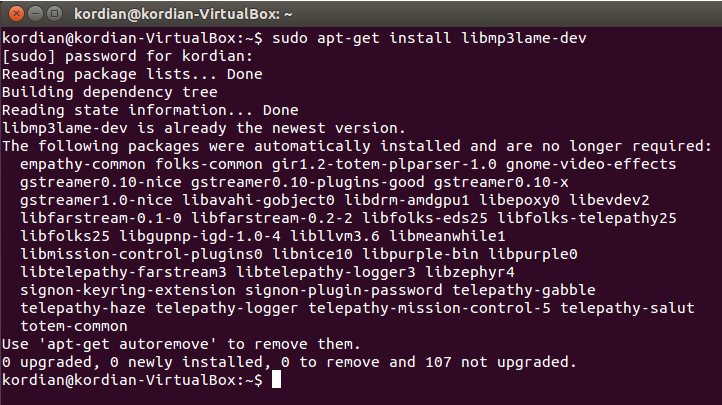
Tak powinien wyglądać poprawnie dodany dysk w VirtualBoxie



A tak włączone Ubuntu ma tym dysku



Wpisujemy w terminal komendę : sudo apt-get install libmp3lame-dev



Dodajemy przykładowy kod z internetu i próbujemy włączyć (sprawdzić czy działa)

**#include <stdio.h>  
#include <lame/lame.h>  
  
int main(void)  
{  
 int read, write;  
 FILE \*pcm = fopen("file.pcm", "rb");  
 FILE \*mp3 = fopen("file.mp3", "wb");  
 const int PCM\_SIZE = 8192;  
 const int MP3\_SIZE = 8192;  
  
 short int pcm\_buffer[PCM\_SIZE\*2];  
 unsigned char mp3\_buffer[MP3\_SIZE];  
 lame\_t lame = lame\_init();  
 lame\_set\_in\_samplerate(lame, 44100);  
 lame\_set\_VBR(lame, vbr\_default);  
 lame\_init\_params(lame);  
  
 do {  
 read = fread(pcm\_buffer, 2\*sizeof(short int), PCM\_SIZE, pcm);  
 if (read == 0)  
 write = lame\_encode\_flush(lame, mp3\_buffer, MP3\_SIZE);  
 else  
 write = lame\_encode\_buffer\_interleaved(lame, pcm\_buffer, read, mp3\_buffer, MP3\_SIZE);  
 fwrite(mp3\_buffer, write, 1, mp3);  
 } while (read != 0);  
  
 lame\_close(lame);  
 fclose(mp3);  
 fclose(pcm);  
  
 return 0;  
}**

Ten kod jest tylko po to, aby sprawdzić czy nie będzie błędów w dodaniu biblioteki lame.h itd.

Otrzymujemy to co chcieliśmy.

tryScreenshot from 2016-05-11 12_41_13.png

Wiemy, że wszystko powinno być poprawnie skonfigurowane. Mamy pewną wiedzę dotyczącą naszego projektu. Zatem Sprint 1 uznajemy za zakończony sukcesem i możemy przystąpić do Sprintu 2.