
PRÉPARATION DE WINDOWS 7/8 POUR LE TP

...

Pour toute question, veuillez contacter

Juliane U. Klamser

Gulliver, ESPCI

Paris

`Juliane.Klamser@espci.psl.eu`

April 2, 2020

Contents

1	Remarques générales	2
2	Type de système : 32 bits ou 64 bits ?	2
3	Installation de MSYS2	3
3.1	Installation de la console Unix	3
3.2	Mise à jour de MSYS2	5
3.3	Installation du compilateur en 64 bits	7
4	Éditeur de code	10
4.1	Notepad++	10
4.2	Atom	11
4.3	VIM ou emacs	11
4.4	Code::Blocks	11
5	Lançons votre premier code C.	11
5.1	Naviguer dans la console jusqu'au dossier du code	12
5.2	Compiler et exécuter le code	13
6	Gnuplot	14
7	Test pour le projet final	14
7.1	Comment travailler simultanément avec quelqu'un d'autre sur le même code ?	16

1 Remarques générales

Il existe de nombreux compilateurs C souvent gratuits (même le compilateur de MicroSoft est gratuit), nous vous proposons d'installer GCC le compilateur de GNU parce qu'il existe sur presque toutes les machines et les systèmes d'exploitation. Coder c'est bien, on peut presque tout faire, mais utiliser le travail des autres c'est encore mieux. Au-delà du compilateur, il est très commode de pouvoir bénéficier de très nombreuses librairies que d'autres, beaucoup plus courageux que vous, ont écrites. En fait, ces librairies sont en permanence améliorées et elles dépendent les unes des autres, les recopier et les garder compatibles entre elles est un challenge. Heureusement il existe des logiciels qui vous permettent de conserver et d'étendre une chaîne de compilation et de librairies à jour sans trop souffrir. C'est particulièrement vrai sur Linux où ils se nomment **apt-get** ou **pacman**. Sur Windows c'était beaucoup moins simple, mais il existe maintenant un logiciel qui vous permet de retrouver l'environnement style Linux pour compiler des applications Windows. C'est ce que nous vous proposons d'utiliser. Il s'agit de **MSYS2** qui va installer une console Linux sur votre Windows, cette console sera très similaire à un terminal de Linux elle vous permettra de lancer le logiciel **pacman** qui se connecte à internet et se charge d'installer et surtout de gérer automatiquement les dépendances des logiciels et librairies que vous installerez. Pour plus tard, vous trouverez peut-être ce site utile, où vous pourrez trouver toutes les bibliothèques de **pacman**: <https://packages.msys2.org/search>

Dans ce qui suit, nous allons décrire comment installer un compilateur associé à un jeu de librairies de fonctions. Cet outil vous permettra de « compiler » un programme que vous aurez écrit et qui utilisera éventuellement des librairies faites par d'autres. Ce compilateur est un programme qui fonctionne à la ligne de commande, c'est assez rudimentaire, mais cela permet de faire beaucoup de choses. Vous aurez également besoin d'un éditeur qui permet d'écrire votre programme (comme Word ou LibreOffice vous permettent d'écrire une lettre). Si vous utilisez déjà un éditeur et qu'il est adapté au langage C continuez à l'utiliser. Si non, vous trouverez plus tard (chapitre 4) quelques propositions pour un éditeur de code.

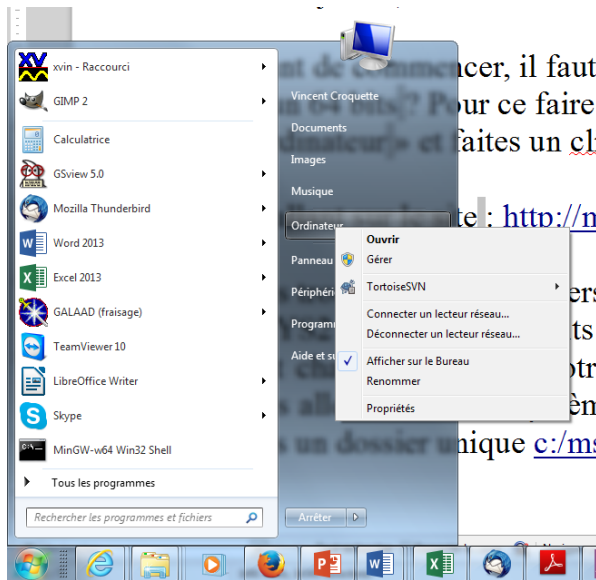
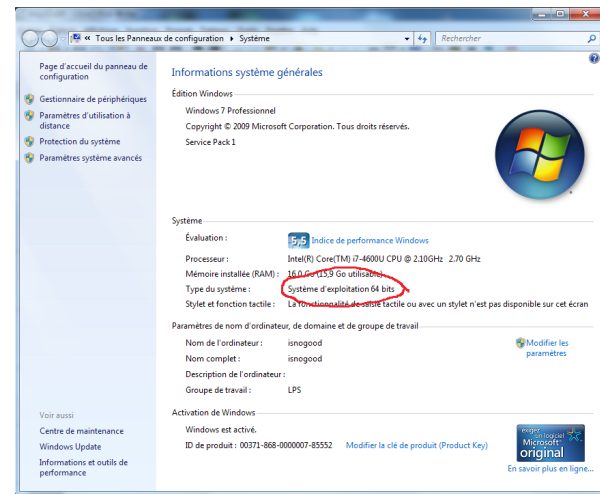
Dans le texte qui suit vous trouverez une description étape par étape de l'installation, en principe si vous suivez cette recette cela devrait marcher. En pratique, vous pouvez disposer d'un ordinateur avec une configuration particulière et il est possible qu'une étape demande un ajustement. Avant de décrire la recette, voici la philosophie de notre installation.

D'abord MSYS2 peut paraître un peu plus compliqué à installer, car ce système est évolutif, vous trouverez d'autres versions de GCC qui s'installent en une opération, mais il est très difficile de rajouter des librairies et encore plus de les mettre à jour. L'installation de MSYS2 se fait en plusieurs étapes :

1. une console du style Unix
2. sa mise à jour
3. l'installation du compilateur 64 bits

2 Type de système : 32 bits ou 64 bits ?

Avant de commencer, il faut déterminer quelle est la version de Windows sur votre PC : est-ce un OS 32 ou un 64 bits ? En principe, de nos jours vous devriez avoir un OS 64 bits, mais ce n'est pas inutile de vérifier. Pour ce faire aller dans le menu « démarrer », placer votre souris au-dessus de « ordinateur » et faites un click **droit** et cliquer sur propriétés, vous devez voir apparaître une fenêtre disant :

(a) « démarrer » → click **droit** sur « ordinateur » → propriétés

(b) l'information requise est marquée en rouge

Figure 1: Trouver votre type de système.

3 Installation de MSYS2

3.1 Installation de la console Unix

Durant toute l'installation, vous devez être connecté à internet. En allant sur le site : <http://msys2.github.io/> vous trouverez des liens vers : MSYS2 en version 32 bits ou MSYS2 en version 64 bits (voir l'image ci-dessous). Suivant votre système, chargez l'un ou l'autre et exécutez cet installateur. Ce faisant vous allez installer le système minimal permettant de démarrer, tous les fichiers utiles seront situés dans un dossier unique c:\msys32 ou c:\msys64

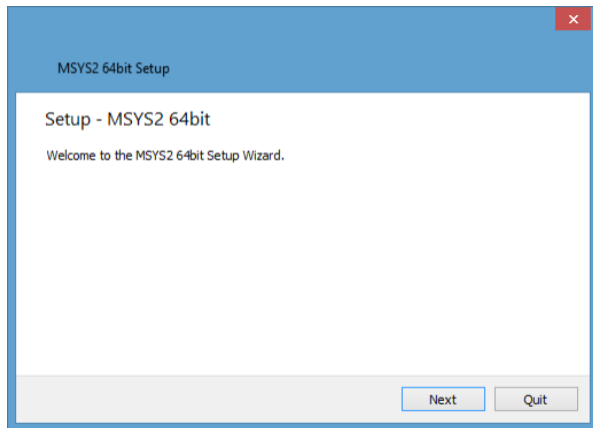


Figure 2: Choisissez la version qui est compatible avec votre ordinateur.

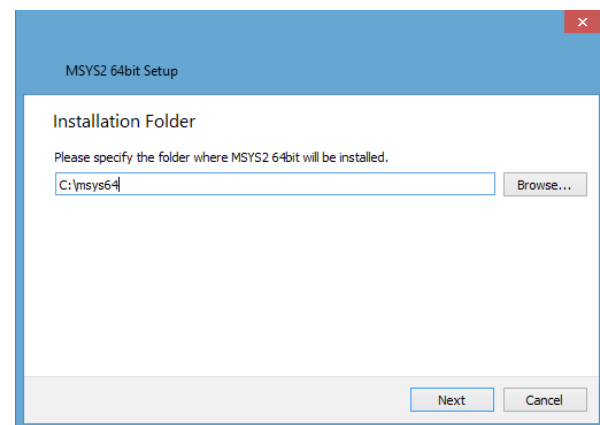
Lancez le processus d'installation.

La description de l'installation est bien faite sur la page de MSYS2. Néanmoins, vous trouverez peut-être utile ce qui suit.

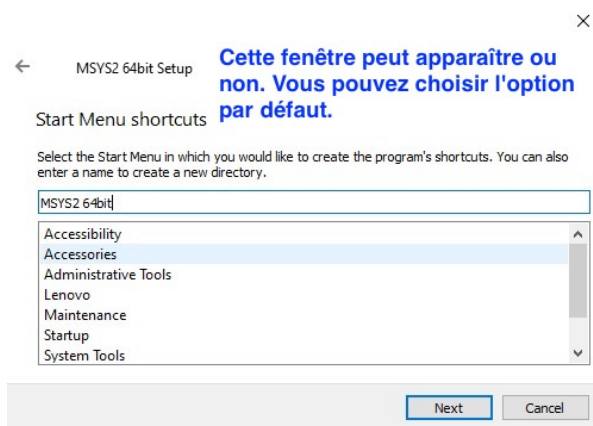
Votre processus d'installation devrait être comparable à la Fig. 3.



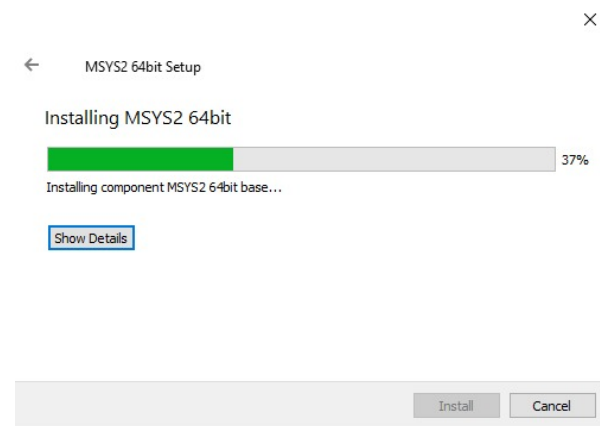
(a) étape 1



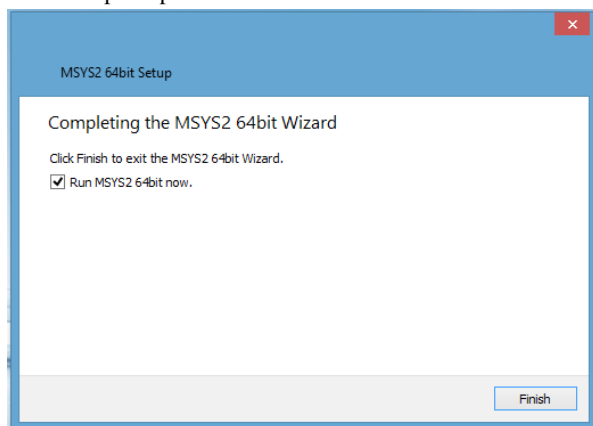
(b) étape 2: Le plus simple est de choisir les options d'installation par défaut. Cependant, si vous savez ce que vous faites, vous pouvez choisir le dossier où vous souhaitez installer votre logiciel, par exemple `c:\msys64` si vous avez un système 64 bits `c:\msys32` sinon. Vous pouvez choisir un autre nom ou installer MSYS2 sur un autre disque (ex. `d:\msys64`), mais surtout ne choisissez pas un nom compliqué ou contenant un espace ! Choisissez un endroit où vous avez le droit d'écrire (surtout pas `c:\Program Files`).



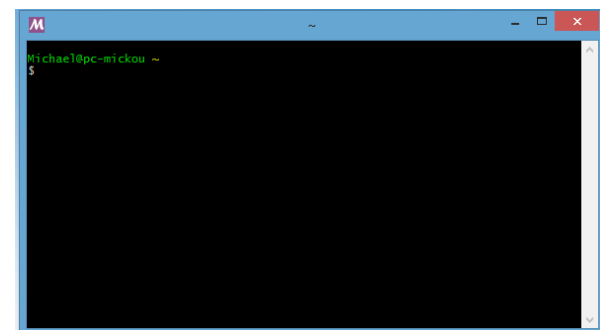
(c) étape 3 : Cette fenêtre peut apparaître ou non. Vous pouvez choisir l'option par défaut.



(d) étape 4 : Attendez que l'installation soit terminée.



(e) étape 5



(f) étape 6: En cliquant sur Finish, vous devriez voir apparaître la console MSYS2 Shell.

Figure 3: Comment l'installation ressemblerait le plus probablement pour vous.

3.2 Mise à jour de MSYS2

He oui, à peine installer il faut mettre à jour, pour cela recopier la ligne de commande ci-dessous (dans le champ noir) et coller la dans la console (click droit, paste). Confirmez avec ENTER !

Ceci est la console MSYS2 Shell

pacman --needed -Sy bash pacman-mirrors msys2-runtime

Notez que vous ne pouvez vous déplacer entre les lettres qu'avec les touches fléchées de votre clavier. Si vous utilisez les flèches haut et bas de votre clavier, vous pouvez naviguer entre les commandes que vous avez utilisées dans le passé dans la console.

Vous devriez voir apparaître quelque chose comme dans la Fig. 4.

```

vincent@PC-bureau MSYS ~
$ pacman --needed -Sy bash pacman-mirrors msys2-runtime
:: Synchronisation des bases de données de paquets...
mingw32                210,7 KiB   356K/s  00:01 [#####] 100%
mingw32.sig             95,0  B   0,00B/s  00:00 [#####] 100%
mingw64                209,2 KiB   236K/s  00:01 [#####] 100%
mingw64.sig             96,0  B   0,00B/s  00:00 [#####] 100%
msys                   125,8 KiB   170K/s  00:01 [#####] 100%
msys.sig                96,0  B   0,00B/s  00:00 [#####] 100%
avertissement[]: bash-4.3.042-1 est à jour -- ignoré
avertissement[]: pacman-4.2.1.6198.439d376-1 est à jour -- ignoré
avertissement[]: pacman-mirrors-20150722-2 est à jour -- ignoré
avertissement[]: msys2-runtime-2.2.1.16655.6ecd782-2 est à jour -- ignoré
il n'y a rien à faire

vincent@PC-bureau MSYS ~
$ |

```

Figure 4: Résultat de `pacman --needed -Sy bash pacman-mirrors msys2-runtime`.

À ce stade, vous devez mettre à jour le logiciel et sa base de données en tapant (il faut être connecté à internet pour que cela marche):

Ceci est la console MSYS2 Shell

pacman -Syu

Vous devriez voir apparaître quelque chose comme dans la Fig. 5.

```
tsadh@DESKTOP-656E1D0 MSYS ~
# pacman -Syu
:: Synchronizing package databases...
mingw32           455.2 KiB   632K/s   00:01 [#####] 100%
mingw32.sig       119.0   B   0.00B/s   00:00 [#####] 100%
mingw64           457.9 KiB   3.41M/s   00:00 [#####] 100%
mingw64.sig       119.0   B   0.00B/s   00:00 [#####] 100%
msys              183.2 KiB   35.8M/s   00:00 [#####] 100%
msys.sig          119.0   B   0.00B/s   00:00 [#####] 100%
:: Starting core system upgrade...
warning: terminate other MSYS2 programs before proceeding
resolving dependencies...
looking for conflicting packages...

Packages (8) bash-4.4.023-2 filesystem-2020.02-2 libzstd-1.4.4-2
             mintty-1-3.1.4-1 msys2-runtime-3.0.7-6 pacman-5.2.1-6
             pacman-mirrors-20200307-1 zstd-1.4.4-2

Total Download Size: 16.51 MiB
Total Installed Size: 65.35 MiB
Net Upgrade Size:     -4.37 MiB

:: Proceed with installation? [Y/n] |
```

Figure 5: Résultat de pacman -Syu.

Tapez Y et appuyez sur ENTER pour continuer. Attendez que l'installation soit terminée.

Note: **Il est possible que le premier essai ne se termine pas avec succès.** Dans ce cas, la console vous indiquera des lignes avec "warning : ...". Lisez le warning et essayez de suivre les instructions. Il vous demandera probablement simplement de fermer la console MSYS2 Shell et de répéter la commande pacman -Syu.

Un résultat réussi devrait ressembler à la Fig. 6 (il n'y a pas de warnings !).

```
(48/60) upgrading gnupg [#####] 100%
==> Appending keys from msys2.gpg...
==> Locally signing trusted keys in keyring...
-> Locally signing key D55E7A6D7CE98A1587C0ACACF40D263ECA25678A...
-> Locally signing key 123D4D51A1793859C28E9168B8E514E53E0D0813...
-> Locally signing key 8918CF33032848F90CC0A3CA9F418C233E652008...
-> Locally signing key 9DD0D4217D75A33B896159E6DA7EF2ABAEAA755C...
==> Importing owner trust values...
==> Updating trust database...
gpg: no need for a trustdb check
(49/60) upgrading libarchive [#####] 100%
(50/60) upgrading libpgpme [#####] 100%
(51/60) upgrading libpcre16 [#####] 100%
(52/60) upgrading libpcre2_8 [#####] 100%
(53/60) upgrading libpcre32 [#####] 100%
(54/60) upgrading libpcrecpp [#####] 100%
(55/60) upgrading libpcreposix [#####] 100%
(56/60) upgrading libutil-linux [#####] 100%
(57/60) upgrading pcre [#####] 100%
(58/60) upgrading xz [#####] 100%
(59/60) upgrading pkgfile [#####] 100%
(60/60) upgrading util-linux [#####] 100%

tsadh@DESKTOP-656E1D0 MSYS ~
#
```

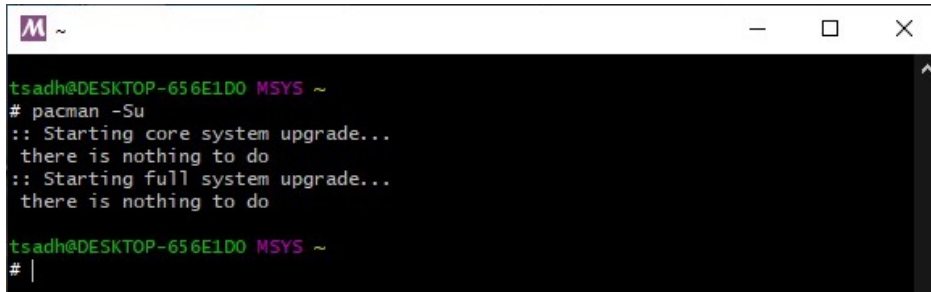
Figure 6: Un résultat réussi de pacman -Syu.

Pour tester si votre installation est réussie. Tapez

Ceci est la console MSYS2 Shell

pacman -Su

Vous devriez obtenir un résultat comme indiqué dans la Fig. 7.



```

tsadh@DESKTOP-656E1D0 MSYS ~
# pacman -Su
:: Starting core system upgrade...
there is nothing to do
:: Starting full system upgrade...
there is nothing to do

tsadh@DESKTOP-656E1D0 MSYS ~
# |

```

Figure 7: Un résultat réussi de pacman -Su.

Vous avez mis à jour la base des paquets de logiciels de votre installation.

À ce stade vous avez installé le logiciel MSYS2 qui contient **3 consoles** et tout ce qu'il faut pour installer et mettre à jour d'autres «paquets» c'est-à-dire des ensembles de logiciels et de bibliothèques. Si vous cliquez sur «démarrer → tous mes programmes» vous allez voir apparaître ceci (Fig. 8):

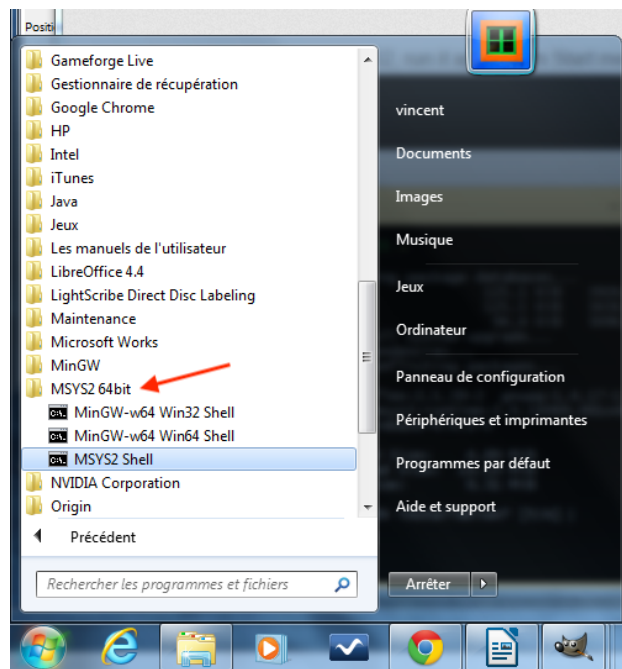


Figure 8: Où trouver les consoles.

Ce sont les trois consoles (Shell en anglais) dont nous venons de parler, en cliquant sur leur nom elles se lancent.

La console **MSYS2 Shell** permet de procéder aux installations et aux mises à jour de tous les paquets de logiciels. Ceux-ci sont de deux types 32 bits et 64 bits.

La console **MinGW-w64 Win32 Shell** permet de travailler avec tous les paquets en 32bits.

La console **MinGW-w64 Win64 Shell** permet de travailler avec tous les paquets en 64bits.

3.3 Installation du compilateur en 64 bits

Vous pouvez maintenant installer le compilateur. L'outil MSYS2 permet de travailler en 64 bits.

Sur une machine 32 bits vous pourrez compiler et exécuter du 32bits, vous pourrez compiler des applications 64 bits, mais elles ne tourneront pas sur votre machine 32 bits.

Avec une machine 64 bits, vous pourrez compiler et exécuter des applications 32 bits et 64 bits.

Nous proposons de travailler avec des applications 64 bits. MSYS2 vous propose deux terminaux différents: **MinGW-w64 Win32 Shell** et **MinGW-w64 Win64 Shell**, qui vous permettent de compiler des applications 32 bits et 64 bits. **Nous utiliserons MinGW-w64 Win64 Shell.**

Mais si vous avez une machine 32 bits, utilisez le compilateur et la Shell en 32 bits.

Quelques rudiments de l'utilisation de la ligne de commande dans une console.

Dans la console vous tapez des commandes suivies d'options éventuelles et vous exécutez cette commande en pressant ENTER:

ls par exemple va lister les fichiers dans le dossier courant en donnant uniquement leurs noms.

ls -l va lister les fichiers avec plus d'informations.

pacman et la commande qui est au cœur du système MSYS2, elle vous permet de rechercher des logiciels puis de les installer en installant également toutes les dépendances. Par la suite elle vous permettra de les mettre à jour. Ici, vous pouvez rechercher des logiciels/paquets <https://packages.msys2.org/search>.

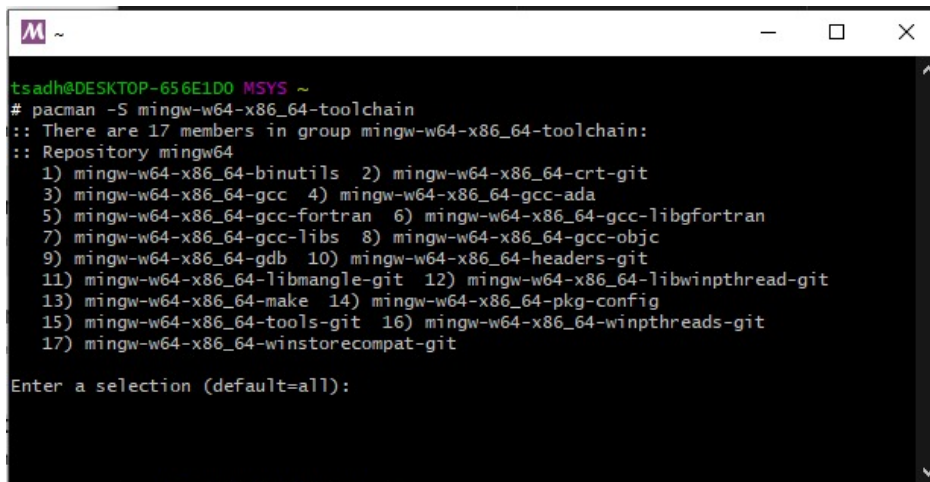
MSYS2 possède un défaut, les noms des paquets sont assez cryptiques, ceci provient du fait qu'ils existent en plusieurs versions (32 et 64 bits), mais aussi que les paquets sont souvent regroupés pour une action particulière.

Pour installer le compilateur GCC en 64 bits, copiez la commande suivante dans la console MSYS2 Shell utilisé précédemment et confirmez avec ENTER

Ceci est la console MSYS2 Shell

pacman -S mingw-w64-x86_64-toolchain

Le résultat devrait ressembler à ceci (Fig. 9). Confirmez avec ENTER pour télécharger tous les paquets par défaut. Attendez ensuite que le téléchargement se termine.

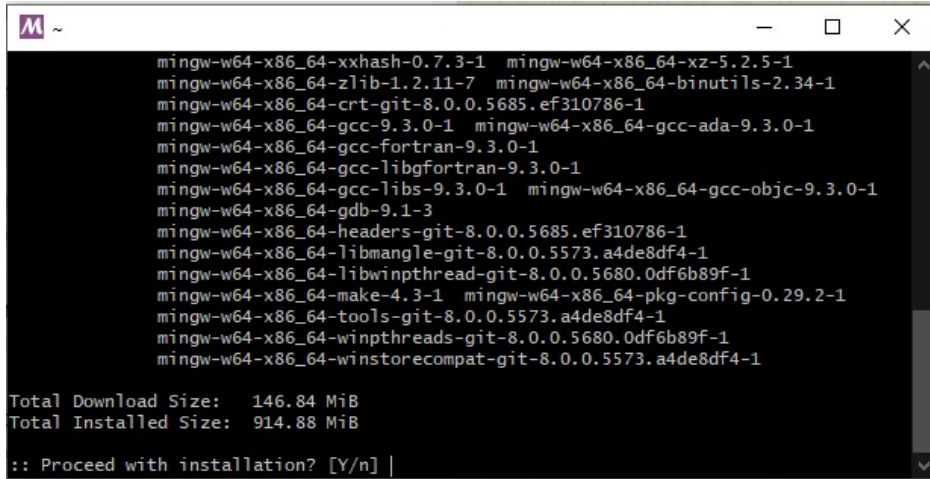


```
tsadh@DESKTOP-656E1D0 MSYS ~
# pacman -S mingw-w64-x86_64-toolchain
:: There are 17 members in group mingw-w64-x86_64-toolchain:
:: Repository mingw64
 1) mingw-w64-x86_64-binutils 2) mingw-w64-x86_64-crt-git
 3) mingw-w64-x86_64-gcc 4) mingw-w64-x86_64-gcc-ada
 5) mingw-w64-x86_64-gcc-fortran 6) mingw-w64-x86_64-gcc-libgfortran
 7) mingw-w64-x86_64-gcc-libs 8) mingw-w64-x86_64-gcc-objc
 9) mingw-w64-x86_64-gdb 10) mingw-w64-x86_64-headers-git
11) mingw-w64-x86_64-libmangle-git 12) mingw-w64-x86_64-libwinpthread-git
13) mingw-w64-x86_64-make 14) mingw-w64-x86_64-pkg-config
15) mingw-w64-x86_64-tools-git 16) mingw-w64-x86_64-winthreads-git
17) mingw-w64-x86_64-winstorecompat-git

Enter a selection (default=all):
```

Figure 9: Un résultat réussi de `pacman -S mingw-w64-x86_64-toolchain`.

Lorsque le téléchargement est terminé, vous devriez voir quelque chose comme dans l'image ci-dessous (Fig. 10). Appuyez sur la touche Y pour procéder à l'installation et attendez que l'installation soit terminée.



```

mingw-w64-x86_64-xxhash-0.7.3-1 mingw-w64-x86_64-xz-5.2.5-1
mingw-w64-x86_64-zlib-1.2.11-7 mingw-w64-x86_64-binutils-2.34-1
mingw-w64-x86_64-crt-git-8.0.0.5685.ef310786-1
mingw-w64-x86_64-gcc-9.3.0-1 mingw-w64-x86_64-gcc-ada-9.3.0-1
mingw-w64-x86_64-gcc-fortran-9.3.0-1
mingw-w64-x86_64-gcc-libgfortran-9.3.0-1
mingw-w64-x86_64-gcc-libs-9.3.0-1 mingw-w64-x86_64-gcc-objc-9.3.0-1
mingw-w64-x86_64-gdb-9.1-3
mingw-w64-x86_64-headers-git-8.0.0.5685.ef310786-1
mingw-w64-x86_64-libmangle-git-8.0.0.5573.a4de8df4-1
mingw-w64-x86_64-libwinpthread-git-8.0.0.5680.0df6b89f-1
mingw-w64-x86_64-make-4.3-1 mingw-w64-x86_64-pkg-config-0.29.2-1
mingw-w64-x86_64-tools-git-8.0.0.5573.a4de8df4-1
mingw-w64-x86_64-winpthreads-git-8.0.0.5680.0df6b89f-1
mingw-w64-x86_64-winstorecompat-git-8.0.0.5573.a4de8df4-1

Total Download Size: 146.84 MiB
Total Installed Size: 914.88 MiB

:: Proceed with installation? [Y/n]

```

Figure 10: Appuyez sur la touche Y pour procéder à l'installation.

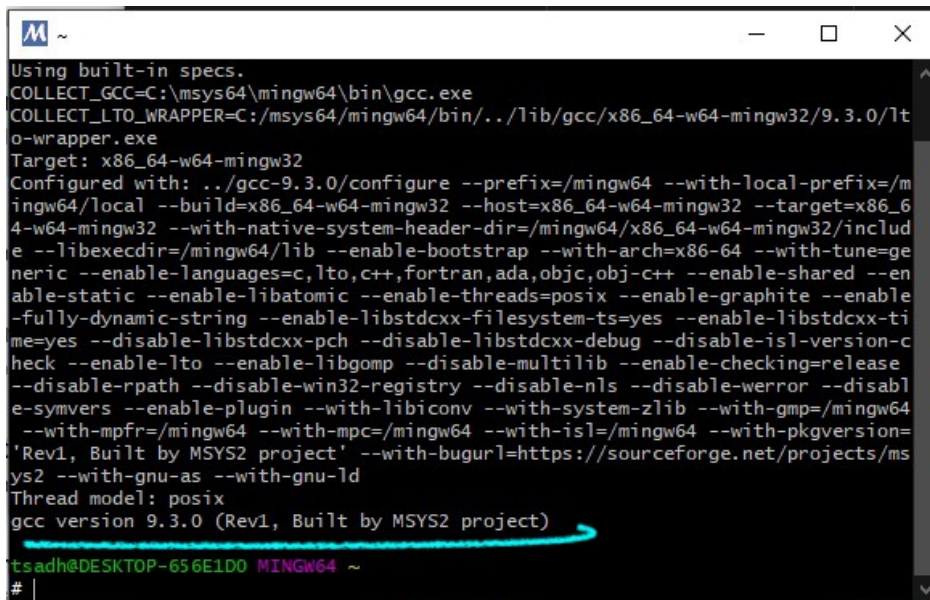
Le processus va prendre un peu de temps, l'ordre en question installe tous les paquets utiles à la compilation des programmes ce qui correspond au chargement de 105Mo soit 609,9 Mo installé au final sur votre PC. Si le processus se passe mal, vous pouvez arrêter la console, la relancer et relancer l'ordre d'installation.

À la fin vous avez installé le compilateur. Pour vérifier que l'installation est bonne, lancer la console **MinGW-w64 Win64 Shell**, une nouvelle fenêtre devrait apparaître tapez la commande

Ceci est la console MinGW-w64 Win64 Shell

gcc -v

(ce qui veut dire lance le compilateur gcc qui m'indique son numéro de version). Vous devriez obtenir un résultat comme dans la Fig. 11. Cela veut dire que vous avez la version 9.3.0 de gcc et que tout va bien. Si vous avez quelque chose comme **gcc command not found**, c'est que l'installation n'a pas marché et il faut la reprendre.



```

Using built-in specs.
COLLECT_GCC=C:\msys64\mingw64\bin\gcc.exe
COLLECT_LTO_WRAPPER=C:/msys64/mingw64/bin/./lib/gcc/x86_64-w64-mingw32/9.3.0/lto-wrapper.exe
Target: x86_64-w64-mingw32
Configured with: ../gcc-9.3.0/configure --prefix=/mingw64 --with-local-prefix=/mingw64/local --build=x86_64-w64-mingw32 --host=x86_64-w64-mingw32 --target=x86_64-w64-mingw32 --with-native-system-header-dir=/mingw64/x86_64-w64-mingw32/include --libexecdir=/mingw64/lib --enable-bootstrap --with-arch=x86_64 --with-tune=generic --enable-languages=c,lto,c++,fortran,ada,objc,obj-c++ --enable-shared --enable-static --enable-libatomic --enable-threads=posix --enable-graphite --enable-fully-dynamic-string --enable-libstdcxx-filesystem-ts=yes --enable-libstdcxx-time=yes --disable-libstdcxx-pch --disable-libstdcxx-debug --disable-lto --enable-lto --enable-libgomp --disable-multilib --enable-checking=release --disable-rpath --disable-win32-registry --disable-nls --disable-werror --disable-symvers --enable-plugin --with-libiconv --with-system-zlib --with-gmp=/mingw64 --with-mpfr=/mingw64 --with-mpc=/mingw64 --with-isl=/mingw64 --with-pkgversion='Rev1, Built by MSYS2 project' --with-bugurl=https://sourceforge.net/projects/msys2 --with-gnu-as --with-gnu-ld
Thread model: posix
gcc version 9.3.0 (Rev1, Built by MSYS2 project)
tsadh@DESKTOP-656E1D0 MINGW64 ~
#

```

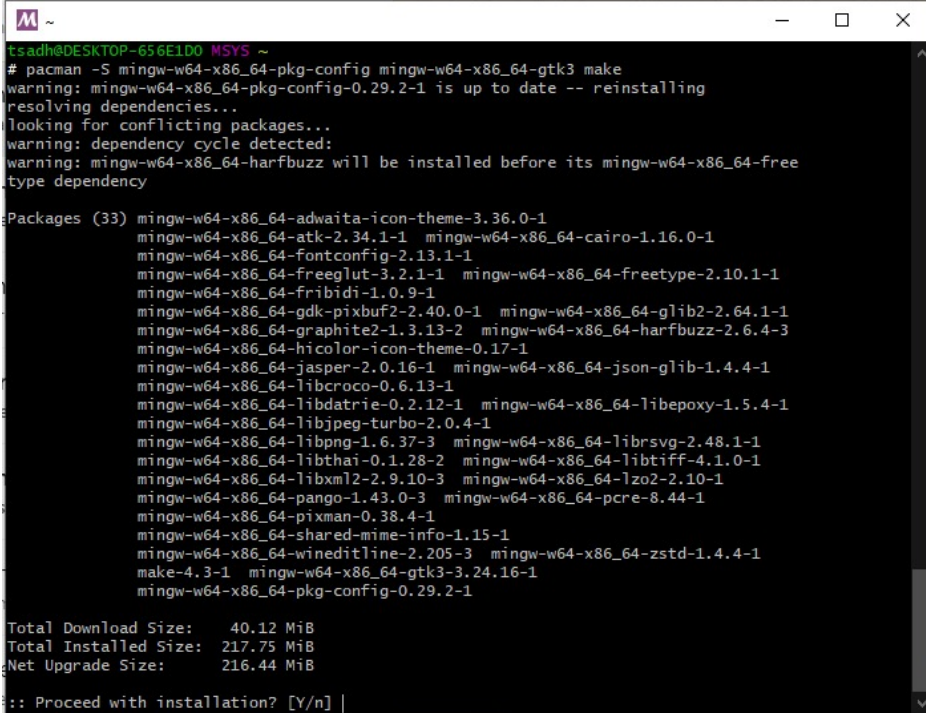
Figure 11: Resultat de gcc -v.

Quand vous avez un compilateur qui marche vous pouvez installer les autres librairies en tapant dans la console **MSYS2 Shell** (attention, changement de console):

Ceci est la console MSYS2 Shell

pacman -S mingw-w64-x86_64-pkg-config mingw-w64-x86_64-gtk3 make

Le résultat devrait ressembler à la figure 12. Là encore, vous devez confirmer le processus d'installation avec Y. Attendez que l'installation soit terminée.



```
tsadh@DESKTOP-656E1D0 MSYS ~
# pacman -S mingw-w64-x86_64-pkg-config mingw-w64-x86_64-gtk3 make
warning: mingw-w64-x86_64-pkg-config-0.29.2-1 is up to date -- reinstalling
resolving dependencies...
looking for conflicting packages...
warning: dependency cycle detected:
warning: mingw-w64-x86_64-harfbuzz will be installed before its mingw-w64-x86_64-free
type dependency

Packages (33) mingw-w64-x86_64-adwaita-icon-theme-3.36.0-1
mingw-w64-x86_64-atk-2.34.1-1 mingw-w64-x86_64-cairo-1.16.0-1
mingw-w64-x86_64-fontconfig-2.13.1-1
mingw-w64-x86_64-freetype-2.10.1-1 mingw-w64-x86_64-fribidi-1.0.9-1
mingw-w64-x86_64-gdk-pixbuf2-2.40.0-1 mingw-w64-x86_64-glib2-2.64.1-1
mingw-w64-x86_64-graphite2-1.3.13-2 mingw-w64-x86_64-harfbuzz-2.6.4-3
mingw-w64-x86_64-hicolor-icon-theme-0.17-1
mingw-w64-x86_64-jasper-2.0.16-1 mingw-w64-x86_64-json-glib-1.4.4-1
mingw-w64-x86_64-libcroco-0.6.13-1
mingw-w64-x86_64-libdatrie-0.2.12-1 mingw-w64-x86_64-libepoxy-1.5.4-1
mingw-w64-x86_64-libjpeg-turbo-2.0.4-1
mingw-w64-x86_64-libpng-1.6.37-3 mingw-w64-x86_64-librsvg-2.48.1-1
mingw-w64-x86_64-libthai-0.1.28-2 mingw-w64-x86_64-libtiff-4.1.0-1
mingw-w64-x86_64-libxml2-2.9.10-3 mingw-w64-x86_64-lzo2-2.10-1
mingw-w64-x86_64-pango-1.43.0-3 mingw-w64-x86_64-pcre-8.44-1
mingw-w64-x86_64-pixman-0.38.4-1
mingw-w64-x86_64-shared-mime-info-1.15-1
mingw-w64-x86_64-wineditline-2.205-3 mingw-w64-x86_64-zstd-1.4.4-1
make-4.3-1 mingw-w64-x86_64-gtk3-3.24.16-1
mingw-w64-x86_64-pkg-config-0.29.2-1

Total Download Size: 40.12 MiB
Total Installed Size: 217.75 MiB
Net Upgrade Size: 216.44 MiB

:: Proceed with installation? [Y/n]
```

Figure 12: Resultat de `pacman -S mingw-w64-x86_64-pkg-config mingw-w64-x86_64-gtk3 make`

4 Éditeur de code

Vous avez plusieurs choix.

4.1 Notepad++

Une façon simple de commencer et d'utiliser un éditeur de texte pour modifier le code source de vos programmes puis de compiler ce code dans la Shell. Vous pouvez installer Notepad++ qui est léger est assez efficace, d'ici <https://notepad-plus-plus.org>. Assurez-vous que vous installez une version compatible avec votre ordinateur.



4.2 Atom

Vous pouvez essayer atom <https://atom.io>

Ce éditeur offre une option qui vous permettent de travailler simultanément sur le même code avec quelqu'un. C'est très utile pour le projet final, lorsque vous travaillez avec un partenaire sur le même code. Cette fonction est appelée "Teletype".

4.3 VIM ou emacs

Il existe des éditeurs plus compliqués comme VIM. Ces éditeurs sont très puissants, ils permettent de compiler le programme dans une de leur fenêtre et il est alors possible de cliquer sur la ligne que le compilateur indique comme erronée pour y arriver directement. Ceci s'avère très pratique. Cependant, elle a besoin d'un peu de formation.

Suivez ce lien pour accéder à un tutoriel vim <https://www.openvim.com>.

Vous obtenez VIM en tapant:

Ceci est la console MSYS2 Shell

```
pacman -S vim
```

Pour emacs tapez

Ceci est la console MSYS2 Shell

```
pacman -S mingw-w64-x86_64-emacs
```

4.4 Code::Blocks

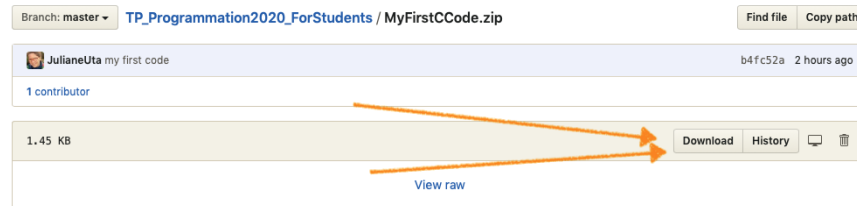
Si vous préférez un environnement intégré de compilation vous pouvez installer l'IDE Code::Blocks qui vous permet d'éviter la ligne de commande si vous êtes allergique et d'écrire et de compiler vos programmes dans un environnement intégré (IDE).

Vous trouverez une description de cet IDE à cette adresse <http://www.codeblocks.org/>

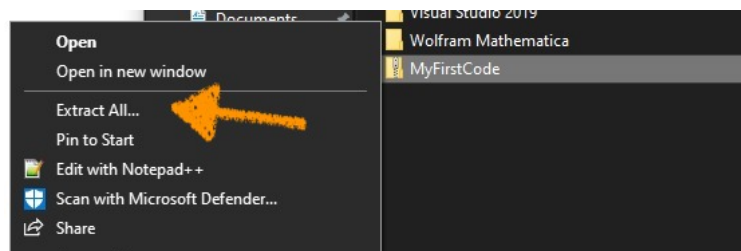
Vous pouvez télécharger le logiciel à cette adresse <http://www.codeblocks.org/downloads>

5 Lançons votre premier code C.

Rendez-vous sur ce site https://github.com/JulianeUta/TP_Programmation2020_ForStudents/blob/master/MyFirstCode.zip et cliquez sur Download. Un dossier compressé avec votre premier code en C sera téléchargé. L'objectif est de compiler le code et de l'exécuter.



Supposons que vous ayez choisi de sauvegarder le dossier compressé dans votre dossier Documents. Ensuite, allez dans votre dossier Documents et faites un clic droit sur le dossier compressé MyFirstCode, puis sélectionnez quelque chose comme "Extraire tout...".



Il devrait y avoir un dossier normal appelé MyFistCode. Allez dans ce dossier. Il y a peut-être un dossier appelé _MACOSX. Vous pouvez supprimer ce dossier _MACOSX (car ce n'est qu'un artefact de la compression des dossiers sur un ordinateur avec un système d'exploitation différent). Ouvrez le fichier main.c avec Notepad++. C'est votre code. Il devrait ressembler à ceci.

```

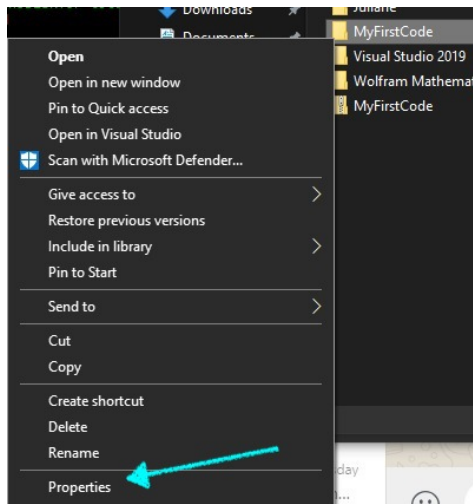
1 #include <stdio.h>
2
3 int main() {
4     // printf() displays the string inside quotation
5     printf("Hello, ESPCI! I am learning C. For a new line I need to type this\n :) \n");
6     return 0;
7 }
8

```

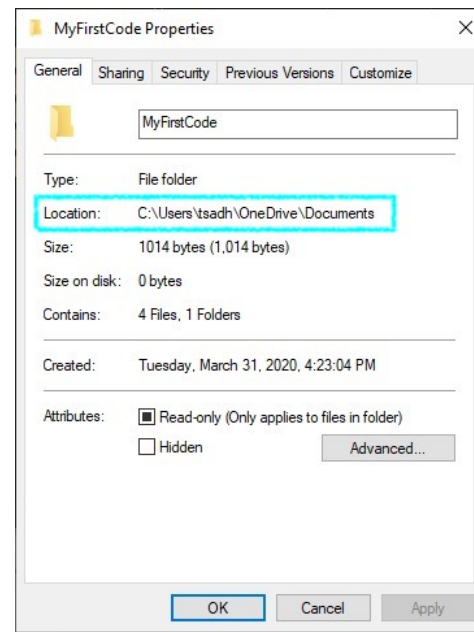
Nous voulons maintenant compiler le code (traduire ces mots écrits en quelque chose qui puisse être interprété par l'ordinateur). **Soyez attentifs, car c'est ce que vous ferez de manière extensive pendant ce TP.**

5.1 Naviguer dans la console jusqu'au dossier du code

Faites un clic droit sur le dossier MyFirstCode et sélectionnez les propriétés. Nous devons connaître la localisation du dossier avec votre code. Dans l'exemple de la Fig. 13, l'adresse dont nous avons besoin se trouve dans le carré bleu (voir Fig. 13b).



(a) clic droit sur MyFirstCode et sélectionnez les propriétés



(b) Adresse du dossier MyFirstCode

Figure 13: Comment trouver l'adresse du dossier.

Nous utilisons les informations contenues dans le carré bleu et les convertissons en une adresse « bash », où les noms de dossiers sont séparés par / et non par \. Dans l'exemple donné, l'adresse « bash » correspondante est (veuillez comparer avec le carré bleu et adapter votre adresse en conséquence) **/c/Users/tsadhu/OneDrive/Documents/MyFirstCode**

Nous utilisons maintenant la console **MinGW-w64 Win64 Shell** et naviguons vers ce dossier en tapant

Ceci est la console MinGW-w64 Win64 Shell

```
cd /c/Users/tsadhu/OneDrive/Documents/MyFirstCode
```

Si vous tapez maintenant

Ceci est la console MinGW-w64 Win64 Shell

```
ls
```

ce qui signifie qu'il faut lister tous les fichiers de ce dossier (c'est bien un L minuscule et non un I majuscule). Vous devriez voir une sortie comme celle-ci.

```
tsadh@DESKTOP-656E1D0 MINGW64 /c/Users/tsadh/OneDrive/Documents/MyFirstCode
# ls
main.c Makefile
```

5.2 Compiler et exécuter le code

Pour compiler le code, que vous avez vu dans votre éditeur du code, tapez

Ceci est la console MinGW-w64 Win64 Shell

```
make
```

Vous devriez voir une sortie comme celle-ci.


```
tsadh@DESKTOP-656E1D0 MINGW64 /c/Users/tsadh/OneDrive/Documents/MyFirstCode
# make
cc -c main.c -Wall -Wextra -Werror -Wshadow -O -pedantic
cc -o main main.o -lm
```

Si vous tapez maintenant

Ceci est la console MinGW-w64 Win64 Shell

ls

Vous devriez voir une sortie comme celle-ci. Il existe deux autres fichiers, main.o et main.exe. main.exe est écrit dans une langue que l'ordinateur peut interpréter et exécuter. C'est ce que l'on appelle l'exécutable.

```
tsadh@DESKTOP-656E1D0 MINGW64 /c/Users/tsadh/OneDrive/Documents/MyFirstCode
# ls
main.c main.exe main.o Makefile
```

Pour exécuter le main.exe, tapez

Ceci est la console MinGW-w64 Win64 Shell

./main.exe

Vous devriez voir une sortie comme celle-ci.

```
tsadh@DESKTOP-656E1D0 MINGW64 /c/Users/tsadh/OneDrive/Documents/MyFirstCode
# ./main.exe
Hello, ESPCI! I am learning C. For a new line I need to type this
:)
```

Il s'agit de la sortie du code. Si vous voyez ce résultat, alors félicitations à vous. Vous êtes maintenant prêt à commencer le TP.

6 Gnuplot

Gnuplot est un logiciel qui sert à produire des représentations graphiques en deux ou trois dimensions de fonctions numériques ou de données. Il sera très utile pour tracer des données avec quelques lignes. Vous pouvez télécharger Gnuplot à partir de ce site. <http://www.gnuplot.info/download.html>

Téléchargez et exécutez l'installateur. Si possible, choisissez les paramètres par défaut. Si vous le souhaitez, vous pouvez choisir le même répertoire que pour l'installation de MSYS2. Une fois l'installation terminée, vous pouvez trouver l'application gnuplot parmi vos programmes.

7 Test pour le projet final

Nous devons tester si votre ordinateur est prêt pour le projet final.

Allez à la page suivante et cliquez sur "Download" :

https://github.com/JulianeUta/TP_Programmation2020_ForStudents/blob/master/MDgtkBasicCodeWin7.zip.

Extrayez le dossier « MDgtkBasicCodeWin7 » (clic droit sur le dossier → extraire tout). Là encore, vous pouvez supprimer le dossier « _MACOSX » à l'intérieur du « MDgtkBasicCodeWin7 ».

Répétez les étapes décrites au chapitre 5.1, pour naviguer la console à l'intérieur du dossier « MDgtkBasicCodeWin7 ». Notez que si vous avez déjà utilisé `cd /path/to/folder/`, vous n'êtes peut-être pas dans le répertoire d'origine. Avec la commande `pwd`, vous pouvez voir dans quel dossier se trouve la console en ce moment.

Ceci est la console MinGW-w64 Win64 Shell

```
pwd
```

Vous devrez peut-être d'abord naviguer vers le dossier d'origine avec la commande

Ceci est la console MinGW-w64 Win64 Shell

```
cd
```

ou vous pouvez aussi revenir en arrière sur une couche du dossier avec la commande

Ceci est la console MinGW-w64 Win64 Shell

```
cd ..
```

et ensuite naviguer de là vers l'intérieur du dossier « MDgtkBasicCodeWin7 ».

Une fois que vous êtes dans le bon dossier (« MDgtkBasicCodeWin7 »), tapez

Ceci est la console MinGW-w64 Win64 Shell

```
ls
```

Votre résultat devrait ressembler à ceci

```

M /c/Users/tsadh/OneDrive/Documents/MDgtkBasicCodeWin7
tsadh@DESKTOP-656E1D0 MINGW64 ~
# cd /c/Users/tsadh/OneDrive/Documents/MDgtkBasicCodeWin7/

tsadh@DESKTOP-656E1D0 MINGW64 /c/Users/tsadh/OneDrive/Documents/MDgtkBasicCodeWin7
# ls
file_picker_gui.c  form_builder.c  graphics.c  gtkcairo.c  Makefile  win_gui.h
file_picker_gui.h  form_builder.h  graphics.h  gtkcairo.h  win_gui.c

tsadh@DESKTOP-656E1D0 MINGW64 /c/Users/tsadh/OneDrive/Documents/MDgtkBasicCodeWin7
#
  
```

Maintenant, compilez le code avec

Ceci est la console MinGW-w64 Win64 Shell

```
make
```

et listez à nouveau tous les fichiers avec

Ceci est la console MinGW-w64 Win64 Shell

```
ls
```

vous devriez voir plusieurs nouveaux fichiers (se terminant par « .o », et `gtkcairo.exe`) comme ici

```
tsadh@DESKTOP-656E1D0 MINGW64 /c/Users/tsadh/OneDrive/Documents/MDgtkBasicCodewin7
# make
gcc -c -Wall -ggdb -I. `pkg-config --libs --cflags gtk+-3.0 gmodule-2.0` file_picker_gui.c -o file_picker_gui.o
gcc -c -Wall -ggdb -I. `pkg-config --libs --cflags gtk+-3.0 gmodule-2.0` form_builder.c -o form_builder.o
gcc -c -Wall -ggdb -I. `pkg-config --libs --cflags gtk+-3.0 gmodule-2.0` win_gui.c -o win_gui.o
gcc -c -Wall -ggdb -I. `pkg-config --libs --cflags gtk+-3.0 gmodule-2.0` graphics.c -o graphics.o
gcc -o gtkcairo.exe file_picker_gui.o form_builder.o win_gui.o graphics.o -Wall -g `pkg-config --libs --cflags gtk+-3.0 gmodule-2.0`

tsadh@DESKTOP-656E1D0 MINGW64 /c/Users/tsadh/OneDrive/Documents/MDgtkBasicCodewin7
# ls
file_picker_gui.c  form_builder.c  graphics.c  gtkcairo.c  Makefile  win_gui.o
file_picker_gui.h  form_builder.h  graphics.h  gtkcairo.exe  win_gui.c
file_picker_gui.o  form_builder.o  graphics.o  gtkcairo.h  win_gui.h

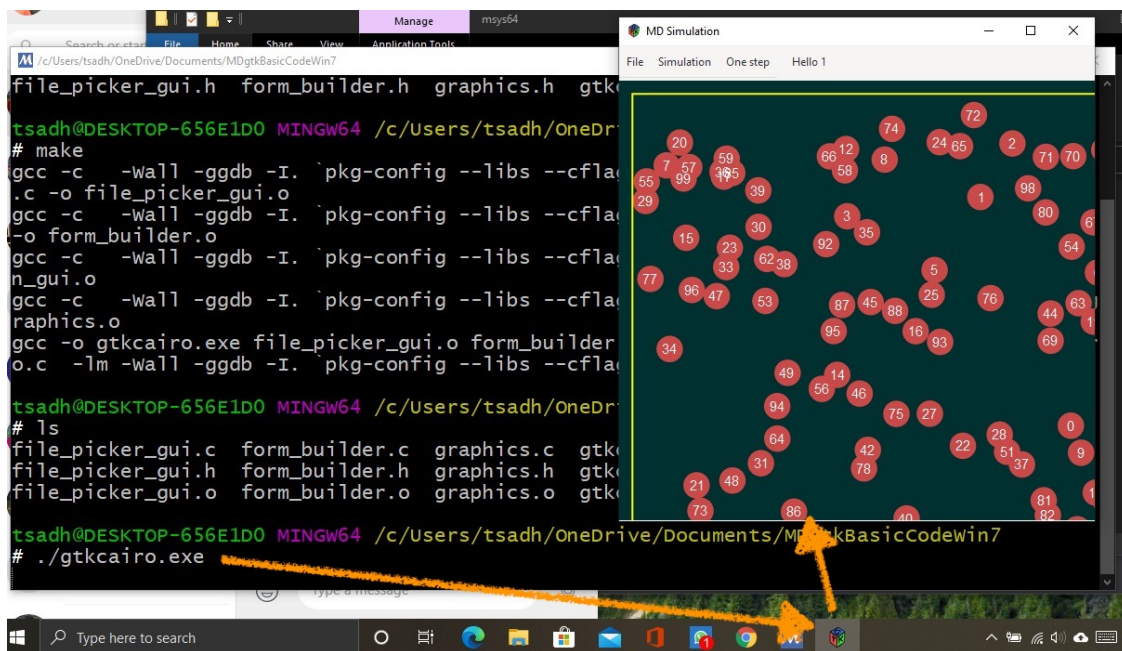
tsadh@DESKTOP-656E1D0 MINGW64 /c/Users/tsadh/OneDrive/Documents/MDgtkBasicCodewin7
#
```

Dans ce cas, l'exécutable est appelé gtkcairo.exe. Nous lançons l'exécutable avec la commande suivante

Ceci est la console MinGW-w64 Win64 Shell

./gtkcairo.exe

Une fenêtre gtk devrait s'ouvrir comme ici



Vous devriez voir un carré vert avec des particules rouges à l'intérieur.

Si vous ne voyez pas la fenêtre gtk avec les cercles rouges, contactez juliane.klamser@espci.psl.eu.

7.1 Comment travailler simultanément avec quelqu'un d'autre sur le même code ?

Comme mentionné au chapitre 4.2, vous pouvez utiliser les fonctions intégrées des éditeurs de code Visual Studio Code ou Atom.

Une autre méthode professionnelle, qu'il convient de mentionner, est GitHub. Elle est très puissante et largement utilisée. Cependant, elle nécessite une certaine formation.

La méthode la plus simple est de créer un compte dropbox et d'utiliser dropbox paper. Si vous copiez votre code dans le fichier du dropbox paper, vous et votre partenaire pouvez travailler sur le même code en même temps.