

Installation automatique

1. installation de Linux
2. création du mot de passe root
3. depuis la session locale, en root, exécuter le script

Script d'installation

Ne pas oublier d'installer le dernier driver vidéo Nvidia pour éviter les problèmes de blocage de l'ordinateur.

x

```
# programmes de base
apt-get -y install autofs portmap nis nfs-common gnome-terminal ssh

# acces root ssh
echo 'PermitRootLogin yes' >> /etc/ssh/sshd_config
service ssh restart

# connexion au serveur
cd /etc
printf "me2" > defaultdomain
printf '#
# yp.conf          Configuration file for the ypbind process. You can define
#                  NIS servers manually here if they can t be found by
#                  broadcasting on the local net (which is the default).
#
#                  See the manual page of ypbind for the syntax of this file.
#
# IMPORTANT:       For the "ypserver", use IP addresses, or make sure that
#                  the host is in /etc/hosts. This file is only interpreted
#                  once, and if DNS isn t reachable yet the ypserver cannot
#                  be resolved and ypbind won t ever bind to the server.

# ypserver ypserver.network.com
domain me2 server 132.207.39.99
#' >yp.conf
printf '# /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch functionality.
# If you have the "glibc-doc-reference" and "info" packages installed, try:
# "info libc "Name Service Switch"' for information about this file.

passwd:          compat nis
```

```
group:          compat nis
shadow:         compat nis
gshadow:        files

hosts:          files mdns4_minimal [NOTFOUND=return] dns myhostname
networks:       files

protocols:      db files
services:       db files
ethers:         db files
rpc:            db files

netgroup:       nis
automount:      nis
```

```
##' > nsswitch.conf
service autofs stop
service ypbind stop
service portmap stop
systemctl add-wants multi-user.target rpcbind.service
service portmap start
service ypbind start
service autofs start
systemctl enable ypbind
systemctl enable autofs
```

```
# interface de connexion
cd /etc/lightdm/lightdm.conf.d/
printf '[SeatDefaults]
user-session=cinnamon
greeter-show-manual-login = true
greeter-hide-users = true
allow-guest = false
```

```
##' > 70-linuxmint.conf
```

```
# activation de xrdp
apt-get install xrdp net-tools
systemctl enable xrdp
```

```
# installation des logiciels requis
```

```
# Mint 21:
```

```
apt-get -y install gedit geany geany-plugin* meld fail2ban cifs-utils nfs-
common libwx-perl libwxbase3.0-dev libboost-dev libuser apache2 fail2ban htop
filezilla rsync members nfs-common sshpass git gfortran gcc g++ libreoffice
imagemagick emacs gummi liblapack-dev libarpack2-dev texlive-full inkscape
```

```
gitg thunar giggle lyx csh nemo sshpass ssh clusterssh gitk blender kwrite  
gimp guake guake-indicator evince gedit libboost-dev libblas-dev liblapack-dev  
libboost-dev libblas-dev liblapack-dev okular rdesktop biber screen jabref  
myspell-fr-gut pavucontrol pdfgrep pdftk-java texstudio pdftk xournal
```

```
# Mint 20:
```

```
# apt-get -y install gedit geany geany-plugin* meld fail2ban cifs-utils nfs-  
common libwx-perl libwxbase3.0-dev libboost-dev libuser libapache2-mod-wsgi  
apache2 fail2ban htop filezilla rsync members nfs-common sshpass git gfortran  
gcc g++ libreoffice imagemagick emacs gummi liblapack-dev libarpack2-dev  
texlive-full inkscape gitg thunar giggle lyx csh nemo sshpass ssh clusterssh  
gitk blender kwrite gimp openshot guake guake-indicator evince gedit libboost-  
dev libblas-dev liblapack-dev libboost-dev libblas-dev liblapack-dev okular  
rdesktop biber screen jabref myspell-fr-gut pavucontrol pdfgrep
```

```
# thèmes Geany
```

```
cp /apps/hephaistos/logiciels/linux/geany/*.conf  
/usr/share/geany/colorschemes/;chmod u=rw  
/usr/share/geany/colorschemes/*.conf;chmod g=r  
/usr/share/geany/colorschemes/*.conf;chmod o=r  
/usr/share/geany/colorschemes/*.conf;
```

```
#Brave
```

```
apt install curl  
curl -fsSLo /usr/share/keyrings/brave-browser-archive-keyring.gpg  
https://brave-browser-apt-release.s3.brave.com/brave-browser-archive-keyring.g  
pg  
echo "deb [signed-by=/usr/share/keyrings/brave-browser-archive-keyring.gpg]  
https://brave-browser-apt-release.s3.brave.com/ stable main"|sudo tee  
/etc/apt/sources.list.d/brave-browser-release.list  
apt update  
apt install brave-browser
```

```
# suppression de Firefox
```

```
apt remove firefox
```

```
# ajout du conteneur Singularity (Salome 2022)
```

```
wget  
https://github.com/sylabs/singularity/releases/download/v3.9.7/singularity-ce_  
3.9.7-bionic_amd64.deb  
sudo apt install ./singularity-ce_3.9.7-bionic_amd64.deb
```

```
apt-get update  
apt-get -y upgrade  
reboot
```

Cette page vous permet de configurer un ordinateur sur le modèle de ceux du laboratoire. La distribution Linux utilisée au laboratoire est [Linux Mint](#). Ce système est basé sur la distribution Ubuntu, trois versions sont rendues disponibles chaque année: aux mois de juin (X.1), d'octobre (X.2) et décembre (X.3).

Liste des versions validées dans le cadre de la procédure décrite sur cette page:

- **Mint 21** ✓ ([oper.](#))
- **Mint 20** ✓ ([oper.](#))
- **Mint 19.3** ✓ ([caduque](#))
- **Mint 19.2** ✓ ([caduque](#))
- **Mint 19.1** ✓ ([caduque](#))
- **Mint 18.2** ✓ ([caduque](#))
- **Mint 18.1** ✓ ([caduque](#))

Les sections débutant avec une icône indiquent les étapes spécifiques aux ordinateurs du laboratoire, à ignorer pour une installation sur un ordinateur personnel.

1. Procédure d'installation

1 - préparation du medium d'installation

Le système d'exploitation est installé à partir d'une clé USB sur laquelle est déposée une image .iso (disponible [ici](#)). Un périphérique USB peut être rendu *bootable* grâce à l'utilisation des logiciels

USB Stick Formatter

et

USB Image Writer

disponibles par défaut sous Linux Mint.

Avec certaines versions de Linux Mint, des problèmes d'accès à des périphériques USB peuvent survenir. Une façon de contourner ces problèmes est d'exécuter les programmes mentionnés précédemment en tant que *root*:

```
su root  
mintstick -m format
```

et

```
su root  
mintstick -m iso
```

2 - installation du système d'exploitation

Les paramètres d'installation utilisés sont ceux par défaut, **sans installation des logiciels tiers**, avec un clavier canadien multilingue. Le fuseau horaire est choisi en renseignant dans le champ de la ville Mont - real car Mint ne semble pas connaître Montréal... Notez que pour pouvoir avoir accès à la sélection de Mont - real, il faut être connecté à internet.

Sur un ordinateur portable, je vous recommande de choisir l'option de cryptage du disque dur (incluant l'espace libre).

En cas de pépin

- pour lancer en ligne de commande l'installation de Mint depuis un disque boot:

```
ubiquity
```

- pour afficher depuis un terminal la fenêtre de paramètres Mint:

```
cinnamon-settings
```

3 - mot de passe root

Depuis un compte utilisateur classique:

```
sudo -i
```

puis:

```
passwd
```

4 - configuration réseau pour connexion à Hephaïstos

Spécifique aux ordinateurs du laboratoire.

Paramétrage d'Hephaïstos

Aucune connexion d'une nouvelle machine n'est possible si le fichier

```
/etc/exports
```

d'Hephaïstos n'a pas été modifié en y ajoutant le nom réseau du nouveau PC installé! Il faudra ensuite exécuter la commande:

```
exportfs -r
```

pour que le nouveau PC puisse se connecter.

Connexion au serveur d'authentification

Une fois le système d'exploitation installé, voici les étapes à suivre *en tant que root* pour vous connecter à Héphaïstos:

1. Si /home n'est pas vide, déplacer son contenu dans un répertoire /home_local:

```
mkdir /home_local/  
chmod 700 /home_local  
mv /home/* /home_local/
```

2. installer les paquets requis:

```
apt-get -y install autofs portmap nis nfs-common gnome-terminal ssh
```

Le nom du serveur à renseigner est me2

3. si aucun écran n'est apparu vous demandant le nom du serveur nis, il faut créer un fichier manuellement:

```
nano /etc/defaultdomain
```

et écrire

```
me2
```

```
-> ctrl + x  
-> y  
-> entrée
```

4. ajouter au fichier yp.conf la ligne:

```
nano /etc/yp.conf
```

et écrire

```
domain me2 server 132.207.39.99
```

5. modifier le fichier nsswitch.conf:

```
nano /etc/nsswitch.conf
```

modifier les lignes:

```
-> passwd : compat nis  
-> group  : compat nis  
-> shadow : compat nis
```

et ajouter à la fin de ce fichier:

```
automount: nis
```

6. arrêter manuellement quelques services:

```
service autofs stop  
service ybind stop  
service portmap stop
```

7. apporter un correctif spécifique à Ubuntu pour permettre au PC de démarrer correctement:

```
systemctl add-wants multi-user.target rpcbind.service
```

8. redémarrer les services éteints:

```
service portmap start  
service ybind start  
service autofs start  
systemctl enable ybind  
systemctl enable autofs
```

À ce stade, en se déplaçant dans

```
/apps/hephaistos/
```

on doit voir les répertoires du serveur.

Inscription DNS

Pour les nouveaux PC, ou lorsque le lieu de connexion du PC a changé, envoyer l'adresse MAC de la machine à notre analyste réseau mec-informatique@polymtl.ca pour l'inscription DNS:

ifconfig

5 - configuration de l'interface de connexion

Lightdm (Ubuntu 16/18, Mint 17/18/19)

Il faut éditer le fichier lightdm.conf (valable pour Mint 19):

```
nano /etc/lightdm/lightdm.conf.d/70-linuxmint.conf
```

et y écrire:

```
[SeatDefaults]
user-session=cinnamon
greeter-show-manual-login = true
greeter-hide-users = true
allow-guest = false
```

Activation de Xrdp

Installer Xrdp et net-tools:

```
apt-get install xrdp net-tools
```

puis activer Xrdp:

```
systemctl enable xrdp
```

Attention: des problèmes de prise en compte des claviers canadiens français multilingues ont été notés. Il faut exécuter la commande:

```
setxkbmap
```

en tant que *root* sur chaque PC pour les régler (testé sur coros-05 en janvier 2022).

6 - configuration locale du poste

Le service `logind` sous Mint semble causer un ralentissement de la connexion au serveur d'authentification (incompatibilité avec NIS probablement). Il est présentement expérimenté de désactiver ce service (coros-02, crs-0x, coros-01):


```
systemctl mask systemd-logind.service
```

Il est déconseillé de désactiver ce service sur une machine qui ne serait pas connectée au réseau du laboratoire.

Mise à jour

Mint 20

Si votre ordinateur dispose d'une carte graphique Nvidia, il peut être requis d'installer les [derniers pilotes](#).

```
apt-get update; apt-get -y install nvidia-driver-440
```

Avant d'installer plus de logiciels, faire cette opération de mise à jour:

```
apt-get -y install fail2ban; apt-get update; apt-get -y upgrade
```

Mint 19.1

Pour **Mint 19.1** uniquement, la mise à jour complète du système génère une erreur. Pour la réparer, il faut exécuter la commande:

```
dpkg --configure -a
```

puis relancer la mise à jour:

```
apt-get -y install fail2ban; apt-get update; apt-get -y upgrade
```

Liste de logiciels requis au laboratoire

La commande de base à exécuter est:

Mint 21

```
apt-get -y install gedit geany geany-plugin* meld fail2ban cifs-utils nfs-common libwx-perl libwxbase3.0-dev libboost-dev libuser apache2 fail2ban htop filezilla rsync members nfs-common sshpass git gfortran gcc g++ libreoffice imagemagick emacs gummi liblapack-dev libarpack2-dev texlive-full inkscape gitg thunar giggle lyx csh nemo sshpass ssh clusterssh gitk blender kwrite gimp guake guake-indicator evince gedit libboost-dev libblas-dev liblapack-dev libboost-dev libblas-dev liblapack-dev okular rdesktop biber screen jabref
```

```
myspell-fr-gut pavucontrol pdfgrep pdftk-java texstudio pdftk xournal
```

Mint 20

```
apt-get -y install gedit geany geany-plugin* meld fail2ban cifs-utils nfs-common libwx-perl libwxbase3.0-dev libboost-dev libuser libapache2-mod-wsgi apache2 fail2ban htop filezilla rsync members nfs-common sshpass git gfortran gcc g++ libreoffice imagemagick emacs gummi liblapack-dev libarpack2-dev texlive-full inkscape gitg thunar giggle lyx csh nemo sshpass ssh clusterssh gitk blender kwrite gimp openshot guake guake-indicator evince gedit libboost-dev libblas-dev liblapack-dev libboost-dev libblas-dev liblapack-dev okular rdesktop biber screen jabref myspell-fr-gut pavucontrol pdfgrep
```

Pour un ordinateur personnel, ajouter les composants:

- [zoom](#)
- [vpn poly](#)
- imprimante
- gscan2pdf
- [synology drive client](#)
- ...

Et éventuellement le mount d'un disque local (le plus simple est d'utiliser l'utilitaire disks de Mint après installation), en tant que root:

```
mkdir /local  
mount /dev/sda2 /local
```

Il faut ensuite créer le lien symbolique suivant:

```
sudo ln -s /usr/lib/x86_64-linux-gnu/libffi.so.7 /usr/lib/x86_64-linux-gnu/libffi.so.6
```

Ce dernier lien est requis pour la bonne utilisation du code `opti_aubes` (les instructions Matplotlib qu'il contient cherchent la bibliothèque de programme `libffi.so.6` qui a été remplacée par `libffi.so.7` sous Linux Mint 20.x, voir. [ce forum](#)).

Mint 19.1

Mint 19.1

```
apt-get -y install gedit geany meld fail2ban cifs-utils nfs-common libwx-perl libwxbase3.0-dev libboost-dev libuser libapache2-mod-wsgi apache2 fail2ban htop filezilla rsync members nfs-common sshpass git gfortran gcc g++
```

```
libreoffice imagemagick emacs gummi liblapack-dev libarpack2-dev texlive-full  
inkscape gitg thunar giggle lyx csh nemo sshpass ssh chromium-browser  
clusterssh gitk blender kwrite gimp openshot guake guake-indicator evince  
gedit libboost-dev libblas-dev liblapack-dev libboost-dev libblas-dev  
liblapack-dev okular rdesktop biber screen jabref myspell-fr-gut snapd  
libwxgtk3.0-dev; snap install pycharm-community --classic
```

2. Installation de logiciels et composants

Mint 20.x

Choix de version

- À partir de la version 2021.0, Salome meca ne fait plus appel à des dépendances systèmes, celles-ci sont incluses dans un "Singularity Container" dans le but de faciliter l'installation.
- Le temps d'ouverture de l'application est significativement plus long à cause du chargement du container, ce qui pose problème dans le contexte d'optimisation itératif. Ci-dessous quelques temps d'exécution:
 - module CST: version 2020.0.1 ⇒ 11 sec VS version 2021.0 ⇒ 19 sec
 - ouverture de salome meca en interactif ⇒ 2020.0.1 ⇒ 1.2 sec VS version 2021.0 ⇒ 4.4 sec
- **Par conséquent la version 2020.0.1 de salome meca est considérée dans tout ce qui suit.**

Vérification de l'intégrité du téléchargement

- De fréquentes erreurs de téléchargement à partir du site de salome meca entraînent la corruption des archives et des erreurs d'exécution (type *erreur MD5*). La procédure ci-dessous est recommandée:
- 1. Copier le lien de la version à télécharger à partir de [cette page](#) et faire le téléchargement à partir du terminal:
 - ```
wget
https://code-aster.org/FICHIERS/salome_meca-2020.0.1-1-universal
.tgz
```
- 2. Décompresser le fichier .tar
- 3. Obtenir le hash du fichier avec la commande suivante et le comparer à celui donné sur la page de téléchargement
  - ```
shasum salome_meca-2020.0.1-1-universal.run
```
 - si le hash n'est pas bon recommencer le téléchargement ! (il m'a fallu plusieurs

essais)

Précautions d'installation

- 1. **placer le fichier téléchargé dans un dossier ne contenant PAS d'accent (même sous linux!) avant de lancer l'installation !!**
- 2. lancer l'installation avec:

- `./salome_meca-2020.0.1-1-universal.run`

- 3. bien suivre les messages d'installation en terminal lors de l'installation, en particulier, il se peut que des librairies soient manquantes. J'ai par exemple dû installer :

- `sudo apt-get install net-tools`

- **vérifier la liste des prérequis [ici](#)**
- bug rencontré sous Ubuntu 22 : [application non créée](#) ⇒ nécessite d'utiliser python2 aliasé comme python temporairement
- 4. tenter le lancement de salome meca en interface graphique d'abord. L'exécutable est localisé dans

- `dossier_installation/appli_V2020.0.1_universal_universal/salome`

- 5. tester l'ouverture des différentes modules dans la barre en haut (GEOM, SMESH, AsterStudy)
- Quelques erreur rencontrées et références utiles pouvait aider à l'installation:
 - *OpenGL_Window::CreateWindow: glXCreateContext failed.* ⇒ [forum](#)
 - *Failed to activate module AsterStudy*, problèmes de librairie libffi.so. ⇒ [forum](#)

- **En cas de besoin [vidéo complète de téléchargement et installation de salome meca](#)** (il s'agit d'une version antérieure mais c'est explicatif quand même)

Requis sous Mint 19.x

Voici une [brève procédure](#) (à valider) trouvée sur internet pour installer Salomé. Plusieurs librairies semblent manquantes sur les PCs Mint 19.1 comparativement à Mint 18.2 et cela empêche le lancement de Salomé...

libcui18n.so.55 manquant

Il faut exécuter (en tant que root) les commandes suivantes:

1. téléchargement:

```
wget  
http://security.ubuntu.com/ubuntu/pool/main/i/icu/libicu55_55.1-7_amd64.de  
b
```

2. exécution:

```
dpkg -i libicu55_55.1-7_amd64.deb
```

3. installation:

```
apt-get -f install
```

libpng12.so.0 manquant

Il faut exécuter (en tant que root) les commandes suivantes:

1. téléchargement:

```
wget  
http://security.ubuntu.com/ubuntu/pool/main/libp/libpng/libpng12-0_1.2.54-  
1ubuntu1.1_amd64.deb
```

2. exécution:

```
dpkg -i libpng12-0_1.2.54-1ubuntu1.1_amd64.deb
```

3. installation:

```
apt-get -f install
```

Autres bibliothèques requises

En tant que root:

```
cd /tmp  
wget  
http://security.ubuntu.com/ubuntu/pool/main/i/icu/libicu55_55.1-7ubuntu0.4_amd  
64.deb  
sudo dpkg -i libicu55_55.1-7ubuntu0.4_amd64.deb  
wget  
http://security.ubuntu.com/ubuntu/pool/main/libp/libpng/libpng12-0_1.2.54-1ubu  
ntu1.1_amd64.deb  
sudo dpkg -i libpng12-0_1.2.54-1ubuntu1.1_amd64.deb  
sudo apt-get install libQt5* net-tools libnlopt0
```

code Aster seul

Vérification de l'intégrité du téléchargement

- De fréquentes erreurs de téléchargement à partir du site de code aster entraînent la corruption des archives et des erreurs d'exécution (type *erreur MD5*). La procédure ci-dessous est recommandée:
- 1. Copier le lien de la version à télécharger à partir de [cette page](#) et faire le téléchargement à partir du terminal:

```
◦ wget  
https://www.code-aster.org/FICHIERS/aster-full-src-14.6.0-1.noarch.tar.gz
```

- 2. Décompresser le fichier .tar
- 3. Obtenir le hash du fichier avec la commande suivante et le comparer à celui donné sur la page de téléchargement

```
◦ shasum nom_fichier
```

◦ si le hash n'est pas bon, recommencer le téléchargement !

précautions d'installation

- 1. **placer le fichier téléchargé dans un dossier ne contenant PAS d'accent (même sous linux!) avant de lancer l'installation !!**
 - voir à ce sujet le dernier message de [ce post](#)
 - si ce n'est pas fait, l'erreur suivante pourrait survenir : *error while loading shared libraries: libmed.so.11: cannot open shared object file: No such file or directory*
- 2. Vérifier que tous les prérequis sont bien installés (voir README)
- 3. Lancer l'installation (infos dans le readme), la compilation prend un peu de temps.
 - désactiver les environnements conda pour l'installation (conda deactivate) (*pas trouvé comment le faire marcher avec même en modifiant les options de setup.cfg*)
 - [lien utile](#)
- 4. L'exécutable aster est

```
dossier_aster/bin/as_run
```

. Tester l'installation avec :

```
◦ as_run --version
```

- 5. Faire un calcul test avec :

- `as_run --vers=14.6 --test forma01a`

- Le résultat est disponible dans ce fichier

recap.txt

. Ce n'est pas grave si il y a une "ALARM" due à l'exécution d'une vieille version de code aster.

- **En cas de besoin** [vidéo complète de téléchargement et installation de aster](#)

OpenFoam / Paraview

Installation d'OpenFoam / Paraview

OpenFoam 6

Les commandes ci-dessous sont issues de la [page d'installation](#) du site officiel d'openfoam. Ces commandes sont à exécuter en tant que root.

1. se placer dans un répertoire temporaire:

```
cd /tmp
```

2. téléchargement:

```
sh -c "wget -O - http://dl.openfoam.org/gpg.key | apt-key add -"
```

3. ajout d'un dépôt:

```
add-apt-repository http://dl.openfoam.org/ubuntu
```

4. recherche des mises à jour:

```
apt-get update
```

5. installation:

```
apt-get -y install openfoam6
```

6. inscription dans le .bashrc:

```
echo '# ' >> ~/.bashrc  
echo '# appel openFoam' >> ~/.bashrc  
echo '. /opt/openfoam6/etc/bashrc' >> ~/.bashrc
```

Pour vérifier la bonne installation d'openfoam, il suffit d'exécuter:

```
simpleFoam
```

Paraview

En tant que root:

```
apt-get install paraview  
apt-get install paraview-dev
```

ou, manuellement:

1. lancer Synaptic:

```
synaptic
```

2. rechercher paraview et cocher

```
paraview-dev
```

pour installation

3. sélectionner apply changes pour installer le logiciel

Pour vérifier la bonne installation de paraView, il suffit d'exécuter:

```
paraFoam -builtin
```

thèmes Geany

Spécifique aux ordinateurs du laboratoire.

Exécutez la commande:

```
cp /apps/hephaistos/logiciels/linux/geany/*.conf  
/usr/share/geany/colorschemes/;chmod u=rw  
/usr/share/geany/colorschemes/*.conf;chmod g=r  
/usr/share/geany/colorschemes/*.conf;chmod o=r  
/usr/share/geany/colorschemes/*.conf;
```


PDFTK (inutile sur les systèmes Mint 21 et plus récents)

Au 20 juillet 2019; les liens internet du script ont été mis à jour. Pour pouvoir installer pdftk sur Ubuntu 18.04 ou Mint 19.x, il faut exécuter en tant que root:

```
cd ~
gedit pdftk_install.sh
```

et copier dans ce fichier les lignes suivantes:

[fichier pdftk_install.sh](#)

```
#!/bin/bash
#
# author: abu
# date: May 12 2018
# description: bash script to install pdftk on Ubuntu 18.04 for amd64 machines
#####
#
# change to /tmp directory
cd /tmp
# download packages
# wget
http://mirrors.kernel.org/ubuntu/pool/main/g/gcc-6/libgcj17_6.4.0-8ubuntu1_amd64.deb \
#
http://mirrors.kernel.org/ubuntu/pool/main/g/gcc-defaults/libgcj-common_6.4-3ubuntu1_all.deb \
#
http://mirrors.kernel.org/ubuntu/pool/universe/p/pdftk/pdftk_2.02-4build1_amd64.deb \
#
http://mirrors.kernel.org/ubuntu/pool/universe/p/pdftk/pdftk-dbg_2.02-4build1_amd64.deb
# ajout 20 juillet 2019:
wget
https://launchpad.net/ubuntu/+source/gcc-6/6.4.0-8ubuntu1/+build/13563762/+files/libgcj17_6.4.0-8ubuntu1_amd64.deb\
http://launchpadlibrarian.net/337429932/libgcj-common_6.4-3ubuntu1_all.deb \
https://launchpad.net/ubuntu/+source/pdftk/2.02-4build1/+build/10581759/+files/pdftk_2.02-4build1_amd64.deb \
https://launchpad.net/ubuntu/+source/pdftk/2.02-4build1/+build/10581759/+files/pdftk-dbg_2.02-4build1_amd64.deb \
# fin ajout
echo -e "Packages for pdftk downloaded\n\n"
```

```
# install packages
echo -e "\n\n Installing pdftk: \n\n"
sudo apt-get install ./libgcj17_6.4.0-8ubuntu1_amd64.deb \
./libgcj-common_6.4-3ubuntu1_all.deb \
./pdftk_2.02-4build1_amd64.deb \
./pdftk-dbg_2.02-4build1_amd64.deb
echo -e "\n\n pdftk installed\n"
echo -e "    try it in shell with: > pdftk \n"
# delete deb files in /tmp directory
rm ./libgcj17_6.4.0-8ubuntu1_amd64.deb
rm ./libgcj-common_6.4-3ubuntu1_all.deb
rm ./pdftk_2.02-4build1_amd64.deb
rm ./pdftk-dbg_2.02-4build1_amd64.deb
```

Il faut ensuite rendre le fichier exécutable:

```
chmod 700 pdftk_install.sh
```

puis l'exécuter:

```
./pdftk_install.sh
```

imprimante (salle C-551)

Spécifique aux ordinateurs du laboratoire.

En tant que root:

```
system-config-printer
```

ajouter une imprimante réseau (M402dw - HP).

paramétrage SSH

Il ne faut pas oublier de permettre la connexion SSH pour root en modifiant l'option

```
PermitRootLogin
```

à:

```
PermitRootLogin yes
```

Depuis la version 19 de la distribution Mint, la configuration par défaut de

```
ssh
```

a été modifiée et cela a pour conséquence de créer un délai (de l'ordre de la trentaine de secondes) lors d'une connexion ssh sur une machine du laboratoire. Pour remédier à ce problème, il faut s'assurer que dans le fichier

```
/etc/ssh/sshd_config
```

les options suivantes sont prises en compte :

```
GSSAPIAuthentication no
GSSAPICleanupCredentials yes
UsePAM no
UseDNS no
```

Plus d'informations à ce sujet [ici](#).

Les fichiers ci-dessous doivent être placés dans /etc/ssh/ (il faut supprimer l'extension .txt):

- ssh_config, [ssh_config.txt contenu du fichier](#):

```
# This is the ssh client system-wide configuration file. See
# ssh_config(5) for more information. This file provides defaults for
# users, and the values can be changed in per-user configuration files
# or on the command line.

# Configuration data is parsed as follows:
# 1. command line options
# 2. user-specific file
# 3. system-wide file
# Any configuration value is only changed the first time it is set.
# Thus, host-specific definitions should be at the beginning of the
# configuration file, and defaults at the end.

# Site-wide defaults for some commonly used options. For a comprehensive
# list of available options, their meanings and defaults, please see the
# ssh_config(5) man page.

Host *
# ForwardAgent no
# ForwardX11 no
# ForwardX11Trusted yes
# RhostsRSAAuthentication no
```

```
# RSAAuthentication yes
# PasswordAuthentication yes
# HostbasedAuthentication no
# GSSAPIAuthentication no
# GSSAPIDelegateCredentials no
# GSSAPIKeyExchange no
# GSSAPITrustDNS no
# BatchMode no
# CheckHostIP yes
# AddressFamily any
# ConnectTimeout 0
# StrictHostKeyChecking ask
# IdentityFile ~/.ssh/identity
# IdentityFile ~/.ssh/id_rsa
# IdentityFile ~/.ssh/id_dsa
# IdentityFile ~/.ssh/id_ecdsa
# IdentityFile ~/.ssh/id_ed25519
# Port 22
# Protocol 2
# Cipher 3des
# Ciphers aes128-ctr,aes192-ctr,aes256-ctr,arcfour256,arcfour128,aes128-cbc,3des-cbc
# MACs hmac-md5,hmac-sha1,umac-64@openssh.com,hmac-ripemd160
# EscapeChar ~
# Tunnel no
# TunnelDevice any:any
# PermitLocalCommand no
# VisualHostKey no
# ProxyCommand ssh -q -W %h:%p gateway.example.com
# RekeyLimit 1G 1h
# SendEnv LANG LC_*
# HashKnownHosts yes
# GSSAPIAuthentication yes
# GSSAPIDelegateCredentials no
```

- sshd_config, [sshd_config.txt](#) contenu du fichier:

```
# Package generated configuration file
# See the sshd_config(5) manpage for details

# What ports, IPs and protocols we listen for
Port 22
# Use these options to restrict which interfaces/protocols sshd will bind to
#ListenAddress ::
```

```
#ListenAddress 0.0.0.0
Protocol 2
# HostKeys for protocol version 2
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key
HostKey /etc/ssh/ssh_host_ecdsa_key
HostKey /etc/ssh/ssh_host_ed25519_key
#Privilege Separation is turned on for security
UsePrivilegeSeparation yes

# Lifetime and size of ephemeral version 1 server key
KeyRegenerationInterval 3600
ServerKeyBits 1024

# Logging
SyslogFacility AUTH
LogLevel INFO

# Authentication:
LoginGraceTime 120
PermitRootLogin yes
#prohibit-password
StrictModes yes

RSAAuthentication yes
PubkeyAuthentication yes
#AuthorizedKeysFile %h/.ssh/authorized_keys

# Don't read the user's ~/.rhosts and ~/.shosts files
IgnoreRhosts yes
# For this to work you will also need host keys in /etc/ssh_known_hosts
RhostsRSAAuthentication no
# similar for protocol version 2
HostbasedAuthentication no
# Uncomment if you don't trust ~/.ssh/known_hosts for
RhostsRSAAuthentication
#IgnoreUserKnownHosts yes

# To enable empty passwords, change to yes (NOT RECOMMENDED)
PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
ChallengeResponseAuthentication no

# Change to no to disable tunnelled clear text passwords
#PasswordAuthentication yes
```

```
# Kerberos options
#KerberosAuthentication no
#KerberosGetAFSToken no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes

# GSSAPI options
#GSSAPIAuthentication no
#GSSAPICleanupCredentials yes

X11Forwarding yes
X11DisplayOffset 10
PrintMotd no
PrintLastLog yes
TCPKeepAlive yes
#UseLogin no

#MaxStartups 10:30:60
#Banner /etc/issue.net

# Allow client to pass locale environment variables
AcceptEnv LANG LC_*

Subsystem sftp /usr/lib/openssh/sftp-server

# Set this to 'yes' to enable PAM authentication, account processing,
# and session processing. If this is enabled, PAM authentication will
# be allowed through the ChallengeResponseAuthentication and
# PasswordAuthentication. Depending on your PAM configuration,
# PAM authentication via ChallengeResponseAuthentication may bypass
# the setting of "PermitRootLogin without-password".
# If you just want the PAM account and session checks to run without
# PAM authentication, then enable this but set PasswordAuthentication
# and ChallengeResponseAuthentication to 'no'.
UsePAM no
UseDNS no
```

Orval

Pour pouvoir lancer Orval, installez les composants suivants:

Mint 20

```
apt-get -y install libboost-dev libboost-all-dev python-subprocess32 libboost-iostreams-dev libboost-system-dev libboost-filesystem-dev gnuplot libopenblas-dev
```

Mint 19.1

```
apt-get -y install libboost-dev libboost-all-dev python-systemd python-ioprocess python-subprocess32 libboost-iostreams-dev libboost-system-dev libboost-filesystem-dev gnuplot libopenblas-dev
```

Ansys

Pour pouvoir exécuter Ansys, il faut également exécuter:

```
apt-get install libmotif-dev
```

puis installer

```
libXp.so.6
```

en téléchargeant dans

```
/tmp
```

[ce fichier](#) et en l'installant à l'aide de la commande:

```
dpkg -i /tmp/libxp6_1.0.2-1ubuntu1_amd64.deb
```

La procédure d'installation de Python MKL, qui peut être faite à ce stade de l'installation est décrite sur cette [page wiki](#).

3. Problèmes connus

pas de son pour les utilisateurs LDAP

Problème fort mystérieux qui semble lié à un problème de permission des comptes LDAP. Après plusieurs

tentatives, et deux reboot, il semble que ces commandes résolvent le problème:

```
apt-install alsa-base  
alsa force-reload
```

Il est aussi recommandé d'installer pavucontrol:

```
apt-get install pavucontrol
```

erreur au lancement de Matlab

Attention: une nouvelle bibliothèque est requise sous Mint 20 pour pouvoir exécuter Matlab et éviter une erreur de type **segmentation fault (core dumped)**:

```
apt-get -y install libncurses5
```

Sur les dernières versions de Mint (> 20) et Ubuntu (> 20.04), le message d'erreur suivant apparaît au lancement de Matlab:

```
p106594@x1:~$ matlab  
MATLAB is selecting SOFTWARE_OPENGL rendering.  
Segmentation fault (core dumped)
```

Sur l'ordinateur où le problème est apparu, Matlab est installé localement dans le répertoire:

```
usr/local/MATLAB/R2015b
```

Le problème a été réparé en procédant de la façon suivante:

1. installation du paquet

```
libncurses5
```

(*legacy version*) avec synaptic

2. création des liens symboliques:

```
ln -s lib/x86_64-linux-gnu/libncursesw.so.6  
/usr/local/MATLAB/R2015b/bin/glnxa64/libncursesw.so.5  
ln -s lib/x86_64-linux-gnu/libncurses.so.5  
/usr/local/MATLAB/R2015b/bin/libncursesw.so.5
```


perte automount au redémarrage du PC

Description du problème

Sur certains ordinateurs (notamment Coros-06), il a été observé une perte de connexion automount après le redémarrage de l'ordinateur. Concrètement, ceci signifie que les répertoires

```
/apps/hephaistos/...
```

ne sont pas accessibles, ce qui empêche toute ouverture de session sur l'ordinateur.

L'exécution en tant que root des commandes suivantes permet de rétablir la connexion:

```
service autofs stop
service ypbind stop
service portmap stop
systemctl add-wants multi-user.target rpcbind.service
service portmap start
service ypbind start
service autofs start
```

Il semble que le problème soit lié à un chargement trop rapide de certains services au démarrage de l'ordinateur. Il n'est toutefois pas clair pourquoi ce problème n'est apparu que sur Coros-06...

Résolution

Benoît Charbonneau propose de suivre les étapes suivantes pour remédier au problème:

1. Créer le fichier

```
/etc/systemd/system/network-online.service
```

2. Mettre ce contenu dans le fichier

```
[Unit]
Description=Wait until NM actually online
Requires=NetworkManager-wait-online.service
After=NetworkManager-wait-online.service

[Service]
Type=oneshot
ExecStart=/usr/bin/nm-online -q --timeout=120
RemainAfterExit=yes

[Install]
WantedBy=multi-user.target

Activer le nouveau service
```

```
systemctl daemon-reload
systemctl enable --now network-online.service
```

Éditer le fichier `/lib/systemd/system/autofs.service` par ajouter `network-online.service` dans `Requires` et `After`

```
[Unit]
Description=Automounts filesystems on demand
Requires=network-online.service network.target rpc-statd.service
rpcbind.service
After=network-online.service network.target ypbind.service sssd.service
network-online.target remote-fs.target
Wants=network-online.target

[Service]
ExecStartPre=/bin/sleep 10
Type=forking
PIDFile=/var/run/autofs.pid
EnvironmentFile=-/etc/default/autofs
ExecStart=/usr/sbin/automount $OPTIONS --pid-file /var/run/autofs.pid
ExecReload=/bin/kill -HUP $MAINPID
TimeoutSec=180

[Install]
WantedBy=multi-user.target
```

erreur au lancement d'Aster study

Si le message d'erreur:

```
ImportError: libffi.so.6: cannot open shared object file: No such file or
directory
```

apparaît au lancement d'Aster study, il faut suivre la procédure suivante:

1. téléchargement du fichier:

```
wget
http://mirrors.kernel.org/ubuntu/pool/main/libf/libffi/libffi6_3.2.1-8_amd
64.deb
```

2. installation de la bibliothèque de programmes associée:

```
sudo apt install ./libffi6_3.2.1-8_amd64.deb
```

Document issu de la page wiki:

<https://lava-wiki.meca.polymtl.ca/ressources/linux/utilisation/base/installation/archives/accueil?rev=1707423126>

Dernière mise à jour: **2024/02/08 15:12**