

Rapport Outils Libres

THIAM Amadou Moctar et KNORST Nicolas

Janvier 2022

1 Efficacité environnement de travail

1.1 Exercice 1 :

- Désactivation de notre souris :

xinput -list → permet de lister les périphériques connectés ;
xinput -set-prop "id" "Device Enabled" 0 → pour désactiver la souris en remplaçant *id* par la valeur affichée.

- Tableau de problèmes et de ses correctifs :

Problème	Correctif
Ouvrir le terminal sous Ubuntu	Ctrl + Alt + T
Démarrer Firefox ou Brave depuis le terminal	\$ firefox ou brave-browser
Saisir directement sur la barre d'URL	Ctrl + E
Page précédente / suivante	Alt + ← — Alt + →
Switcher entre les onglets du terminal (raccourcis créer)	Alt + ← — Alt + →

1.2 Exercice 2 :

- Nous avons choisi TypingClub comme site web pour améliorer notre saisie automatique ;
- Nous l'avons choisi parce que :
 - Une interface utilisateur belle et propre ;
 - Possibilité de s'entraîner pas à pas pour progresser à son rythme et avoir un aperçu de ses progrès ;
 - Possibilité de créer un compte et de sauvegarder ses progrès ;
 - Possibilité d'apprendre en fonction de la disposition de son clavier ;
 - Possibilité d'apprendre en mode jeu.
 - Capture de l'interface

jjkkkj kjkkj jkkkj kkjkk jkkkj jkkkj jkkkj jkkkj jkkkj jkkkj



RATATYPECOURSTESTAPPRENDRELES COURS

Flash 16:50

A stylized illustration of a hand holding a stopwatch. The hand is pink and the stopwatch is blue and white. The background is a solid blue color with radiating lines.

Passez un test de dactylographie gratuit

Tapez un court texte pour savoir à quelle vitesse vous pouvez taper, testez votre vitesse de frappe en mpm en français, anglais ou espagnol. Impressionnez amis ou employeurs avec votre propre certificat.

PASSEZ UN TEST DE DACTYLOGRAPHIE

1.3 Exercice 3 :

- Effectuer les tutoriels avec VIM :

```
=====
=  B i e n v e n u e  d a n s  l e  T u t o r i e l  d e  V I M  -  V e r s i o n  1.7  =
=====

Vim est un éditeur très puissant qui a trop de commandes pour pouvoir
toutes les expliquer dans un cours comme celui-ci, qui est conçu pour en
décrire suffisamment afin de vous permettre d'utiliser simplement Vim.

Le temps requis pour suivre ce cours est d'environ 25 à 30 minutes, selon
le temps que vous passerez à expérimenter.

ATTENTION :
Les commandes utilisées dans les leçons modifieront le texte. Faites une
copie de ce fichier afin de vous entraîner dessus (si vous avez lancé
"vintutor" ceci est déjà une copie).

Il est important de garder en tête que ce cours est conçu pour apprendre
par la pratique. Cela signifie que vous devez exécuter les commandes
pour les apprendre correctement. Si vous vous contentez de lire le texte,
vous oublierez les commandes !

Maintenant, vérifiez que votre clavier n'est PAS verrouillé en
majuscules, et appuyez la touche j le nombre de fois suffisant pour
que la Leçon 1.1 remplisse complètement l'écran.

-----
Leçon 1.1 : DÉPLACEMENT DU CURSEUR

** Pour déplacer le curseur, appuyez les touches h,j,k,l comme indiqué. **
      ^
      k
< h   l >   Astuce : La touche h est à gauche et déplace à gauche.
      j       La touche l est à droite et déplace à droite.
      v       La touche j ressemble à une flèche vers le bas.

1. Déplacez le curseur sur l'écran jusqu'à vous sentir à l'aise.
"/tmp/tutorCjH8Ng" 1038 lignes, 39314 caractères
```

- Paramétrage de GNU Readline :
 - `$ set -o vi`
- L'éditeur choisi est Vim
- Vim comme éditeur par défaut : dans le fichier `.bashrc` :
 - `export VISUAL=vim`
 - `export EDITOR="$VISUAL"`

1.4 Exercice 4 :

- Regarder notre historique :
 - `$ history` ou bien `$ cat /home/user/.bash_history`
- Oui il y en a certaines commandes sensibles et la façon de gérer cela est d'empêcher ces commandes de l'historique bash lui-même
- Pour éviter que certaines commandes de l'historique n'apparaissent, dans le fichier `.bashrc`, on ajoute la ligne suivante :
 - `HISTIGNORE = ls cd pwd`

1.5 Exercice 5 :

- Écriture d'une bash fonction `mkcd` :

Dans le fichier `.bashrc`, on ajoute les lignes suivantes :

 - `mkcd() {`
`mkdir "$1" ;`
`cd "$1" ;`
`}`

- Écriture de la fonction `gitemergency` :

```
– gitemergency() {
    git add .;
    git commit -m "@";
    git push origin Head;
}
```

1.6 Exercice 6 :

- `$ touch backup.sh` → création du fichier backup
- `$ chmod +x backup.sh`
- Code du fichier `bash_completion` :

```
if ! shopt -oq posix; then
if [ -f /usr/share/bash-completion/bash_completion ]; then
. /usr/share/bash-completion/bash_completion
elif [ -f /etc/bash_completion ]; then
. /etc/bash_completion
fi
fi
```

- `$ source bash_autocompletion`

1.7 Exercice 7 :

- Installation de ZSH :
 - `(sudo) apt install zsh -y`
- Modification du prompt pour inclure les informations de vagrant :
Les configurations sont à faire dans le fichier `.zshrc`, en spécifiant :
 - `ZSH_THEME=agnoster` → pour spécifier le nom du thème choisi ;
 - `plugin=vagrant-prompt.plugin.zsh` → choisir le nom du
 - Dans le fichier `.oh-my-zsh/themes/agnoster.zsh-theme`, on ajoute les lignes suivantes :

PROMPT='%{\$fg[NCOLOR]%}%B%n%b%{\$reset_color%}:%{\$fg[blue]%}%B%c/%b%{\$reset_color%}
\$(vagrant_prompt_info)\$(svn_prompt_info)\$(git_prompt_info)%(!.#.\$)'

→ modification de variable de prompt ;

- Ajout des variables de couleurs :

```
ZSH_THEME_VAGRANT_PROMPT_PREFIX="%{$fg_bold[blue]%}["
ZSH_THEME_VAGRANT_PROMPT_SUFFIX="%{$fg_bold[blue]%}%{$reset_color%} "
ZSH_THEME_VAGRANT_PROMPT_RUNNING="%{$fg_no_bold[green]%}●"
ZSH_THEME_VAGRANT_PROMPT_POWEROFF="%{$fg_no_bold[red]%}●"
ZSH_THEME_VAGRANT_PROMPT_SUSPENDED="%{$fg_no_bold[yellow]%}●"
ZSH_THEME_VAGRANT_PROMPT_NOT_CREATED="%{$fg_no_bold[white]%}○"
```

Exercice 9 :

- Installation de trois émulateurs de terminaux à savoir guake, roxterm et xterm:
 - *\$ sudo apt-get install -y guake roxterm xterm*
- Nous avons choisi Guake comme émulateur parce qu'il est : léger, rapide, multi-tab et hautement personnalisable.
- Suppression des autres :
 - *\$ sudo apt-get --purge remove -y roxterm xterm*

2 SSH :

2.1 Exercice 1 :

- `$ vagrant up` → Démarrer l'environnement Vagrant :
- `$ ssh bob@srv.local` → connexion avec l'utilisateur bob
- `$ ssh alice@srv.local` → connexion avec l'utilisateur alice
- `$ ssh carole@srv.local` → connexion avec l'utilisateur carole
- Vérification qu'on est sur la Vagrant :

<pre>bob@srv:~\$ hostname && users srv bob bob@srv:~\$</pre>	<pre>alice@srv:~\$ hostname && users srv alice alice@srv:~\$</pre>	<pre>carol@srv:~\$ hostname && users srv carol carol@srv:~\$</pre>
--	--	--

- On remarque que la commande de la connexion SSH sur la machine vagrant et présente

Exercice 2 :

- Création d'une paire de clé privée et public :
 - `$ ssh-keygen -q -t rsa -f my_keys -C " " -N 'passer'`
- Dépôt de la clé publique :
 - `$ ssh-copy-id -i ~/.ssh/my_keys.pub alice@cli.local`

```
amadou@amadou:~/ssh$ ssh-copy-id -i my_keys.pub alice@192.168.56.2
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "my_keys.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is
alice@192.168.56.2's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'alice@192.168.56.2'"
and check to make sure that only the key(s) you wanted were added.

amadou@amadou:~/ssh$ ssh alice@192.168.56.2
Linux cli 4.9.0-12-amd64 #1 SMP Debian 4.9.210-1 (2020-01-20) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
alice@cli:~$
```

- Dépôt manuelle de la clé publique vers bob:
 - `$ scp ~/.ssh/my_key.pub bob@cli.local :~/.ssh/authorized_keys`
 - `$ ssh bob@cli.local`

```

amadou@amadou:~/.ssh$ scp my_keys.pub bob@192.168.56.2:~/.ssh/authorized_keys
bob@192.168.56.2's password:
my_keys.pub
amadou@amadou:~/.ssh$ ssh bob@192.168.56.2
Linux cli 4.9.0-12-amd64 #1 SMP Debian 4.9.210-1 (2020-01-20) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Dec  6 09:17:30 2021 from 192.168.56.1
bob@cli:~$

```

- La passphrase d'une clé privée permet d'ajouter une couche supplémentaire de sécurité en cas de vol ou de divulgation accidentelle de la clé privée.

2.2 Exercice 3 :

- Suppression des adresses ajoutées dans le fichier "known_host"
 - *\$ ssh-keygen -R cli.local*
 - *\$ ssh-keygen -R srv.local*
- Ajout de la clé publique de srv (192.168.56.3)
 - *\$ ssh-keyscan -H 192.168.56.3 > known_hosts*
- – Création du fichier de configuration SSH


```

Host bc
Hostname 192.168.56.2
User bob
Port 22
PreferredAuthentications publickey
IdentityFile ~/.ssh/My_keys.pub

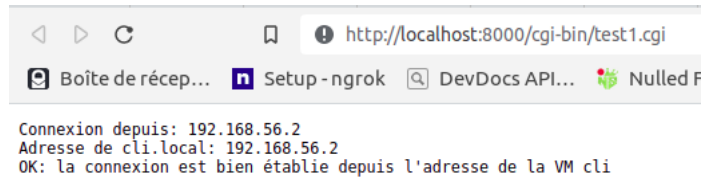
```

2.3 Exercice 3-bis :

- Connexion à la machine cli en utilisant SFTP
 - *\$ sftp alice@cli.local*
 - *\$ put /home/amadou/Bureau/exemple_file.txt* → copie d'un fichier de la machine local vers cli ;
 - *\$ echo "remote file" > remote_file* → création d'un fichier sur cli ;
 - *\$ get remote_file /home/amadou/Bureau/* → copie du fichier vers la machine hôte.
- Connexion à cli avec SSHFS
 - *\$ mkdir ~/mount_rep/* → création du répertoire de montage ;
 - *\$ sshfs alice@cli.local:/home/alice/ ~/mount_rep/* →

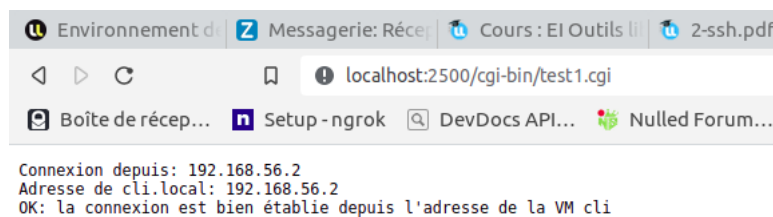
2.4 Exercice 4 :

- Création d'un tunnel SSH avec SRV à travers CLI
 - `$ ssh -N -f -L 8000:srv.local:80 alice@cli.local`



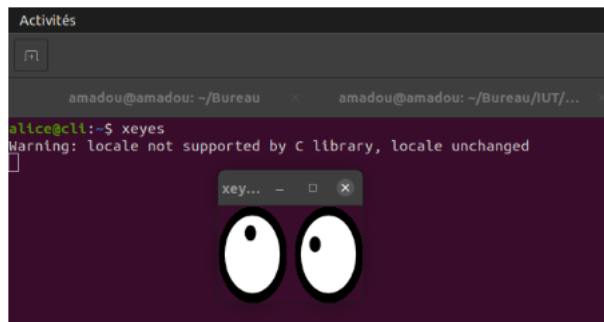
2.5 Exercice 5 :

- Création d'un tunnel SSH entre notre poste et SRV à travers la machine CLI
 - `$ ssh -L 2500:srv.local:80 alice@cli.local`
- Test en ouvrant le navigateur sur notre poste local



2.6 Exercice 6 :

- Connexion à CLI en activant le forwarding X11
 - `$ ssh -X alice@cli.local`
- Lancement d'une application graphique



2.7 Exercice 7 :

- Avec proxyjump :

– `$ ssh -J alice@cli.local bob@srv.local`

```
amadou@amadou: ~/Bureau/IUT/Outils_Libres_Cote_client/SSH/vagrant$ ssh -J alice@cli.local alice@srv.local
alice@srv.local's password:
Linux srv 4.9.0-12-amd64 #1 SMP Debian 4.9.210-1 (2020-01-20) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Dec 13 10:22:27 2021 from 192.168.56.1
alice@srv:~$ who
alice pts/0 Jan 5 21:52 (192.168.56.2)
alice@srv:~$
```

- Avec ProxyCommand :

```
GNU nano 4.8
Host cli.local
Hostname cli.local
User alice

Host srv
Hostname srv.local
User alice
ProxyCommand ssh -W %h:%p cli.local
```

3 GIT :

3.1 Exerice 1 :

- *\$ mkdir git_rep* → création du répertoire de travail
\$ git init → initialisation de git ;
\$ git status

```
amadou@amadou:~/Bureau/IUT/Outils_Libres_Cote_client/SSH/vagrant/git_rep$ git status
Sur la branche master

Aucun commit

Fichiers non suivis:
(utilisez "git add <fichier>..." pour inclure dans ce qui sera validé)
    Vagrantfile
    srv/

aucune modification ajoutée à la validation mais des fichiers non suivis sont présents (utilisez "git add" pour les suivre)
amadou@amadou:~/Bureau/IUT/Outils_Libres_Cote_client/SSH/vagrant/git_rep$
```

- *\$ vagrant up*
\$ vagrant halt → initialisation de git ;
\$ git status ;

```
amadou@amadou:~/Bureau/IUT/Outils_Libres_Cote_client/SSH/vagrant/git_rep$ git status
Sur la branche master

Aucun commit

Fichiers non suivis:
(utilisez "git add <fichier>..." pour inclure dans ce qui sera validé)
    .vagrant/
    Vagrantfile
    srv/

aucune modification ajoutée à la validation mais des fichiers non suivis sont présents (utilisez "git add" pour les suivre)
amadou@amadou:~/Bureau/IUT/Outils_Libres_Cote_client/SSH/vagrant/git_rep$
```

Nous constatons que le dossier *".vagrant"* a été ajouté aux fichiers suivi, alors qu'il ne devrait pas y être. Pour l'enlever il suffit d'ajouter dans le fichier *".gitignore"* la valeur suivant : *".vagrant"*

```
amadou@amadou:~/Bureau/IUT/Outils_Libres_Cote_client/SSH/vagrant/git_rep$ git status
Sur la branche master

Aucun commit

Fichiers non suivis:
(utilisez "git add <fichier>..." pour inclure dans ce qui sera validé)
    Vagrantfile
    srv/

aucune modification ajoutée à la validation mais des fichiers non suivis sont présents (utilisez "git add" pour les suivre)
amadou@amadou:~/Bureau/IUT/Outils_Libres_Cote_client/SSH/vagrant/git_rep$
```

- *\$ git add Vagrantfile srv*

```
amadou@amadou:~/Bureau/IUT/Outils_Libres_Cote_client/SSH/vagrant/git_rep$ git add Vagrantfile
amadou@amadou:~/Bureau/IUT/Outils_Libres_Cote_client/SSH/vagrant/git_rep$ git add srv
amadou@amadou:~/Bureau/IUT/Outils_Libres_Cote_client/SSH/vagrant/git_rep$ git status
Sur la branche master

Aucun commit

Modifications qui seront validées :
(utilisez "git rm --cached <fichier>..." pour désindexer)
    nouveau fichier : Vagrantfile
    nouveau fichier : srv/test1.cgi
    nouveau fichier : srv/test2.cgi

amadou@amadou:~/Bureau/IUT/Outils_Libres_Cote_client/SSH/vagrant/git_rep$
```

- `$ git commit -m "commit_1"`

```
amadou@amadou:~/Bureau/IUT/Outils_Libres_Cote_client/SSH/vagrant/git_rep$ git commit -m "commit_1"
[master (commit racine) bidd695] commit_1
3 files changed, 70 insertions(+)
create mode 100644 Vagrantfile
create mode 100755 srv/test1.cgi
create mode 100755 srv/test2.cgi
```

- `$ git show`

```
amadou@amadou:~/Bureau/IUT/Outils_Libres_Cote_client/SSH/vagrant/git_rep$ git show
commit bidd6954f75d0ee4b33862476e16b60cc5413921 (HEAD -> master)
Author: Amadou THIAM <amadouth@gmail.com>
Date: Thu Jan 13 10:18:09 2022 +0100

    commit_1

diff --git a/Vagrantfile b/Vagrantfile
new file mode 100644
index 0000000..b8f4404
--- /dev/null
+++ b/Vagrantfile
@@ -0,0 +1,44 @@
+Vagrant.configure("2") do |config|
+  config.vm.provider "virtualbox" do |vb|
+    vb.memory = "1024"
+    vb.cpus = 1
+  end
```

3.2 Exercice 2 :

- Création d'une nouvelle branche
 - `$ git checkout -b branche_1`
- Ajout de l'utilisateur Patrick

```
useradd --shell /bin/bash --create-home patrick || true
echo patrick:passer123 | chpasswd
echo alice:1234 | chpasswd
```

- Installation et activation de php dans apache

```
add-apt-repository ppa:ondrej/php
apt update
apt-get install -y php7.4 apache2
a2enmod php7.4
```

- effectuer deux commits distincts

```
amadou@amadou:~/Bureau/IUT/Outils_Libres_Cote_client/SSH/vagrant/git_rep$ git add -p
diff --git a/Vagrantfile b/Vagrantfile
index b8f4404..7a437d7 100644
--- a/Vagrantfile
+++ b/Vagrantfile
@@ -9,9 +9,15 @@ Vagrant.configure("2") do |config|
  set -ex
  apt-get update
  apt-get -y install libnss-mdns
+  add-apt-repository ppa:ondrej/php
+  apt update
+  apt-get install -y php7.4 apache2
+  a2enmod php7.4
  useradd --shell /bin/bash --create-home alice || true
  useradd --shell /bin/bash --create-home bob || true
  useradd --shell /bin/bash --create-home carol || true
+  useradd --shell /bin/bash --create-home patrick || true
+  echo patrick:passer123 | chpasswd
  echo alice:1234 | chpasswd
  echo bob:azerty | chpasswd
  echo carol:secret | chpasswd
(1/1) Indexer cette section [y,n,q,a,d,s,e,?] ?
```

- Revenir à la branche master (main)

```
amadou@amadou:~/Bureau/IUT/Outils_Libres_Cote_client/SSH/vagrant/git_rep$ git branch
* branche_1
  master
amadou@amadou:~/Bureau/IUT/Outils_Libres_Cote_client/SSH/vagrant/git_rep$ git checkout master
M      Vagrantfile
Basculement sur la branche 'master'
amadou@amadou:~/Bureau/IUT/Outils_Libres_Cote_client/SSH/vagrant/git_rep$
```

- L'état de notre working directory

```
amadou@amadou:~/Bureau/IUT/Outils_Libres_Cote_client/SSH/vagrant/git_rep$ git status
Sur la branche master
Modifications qui seront validées :
  (utilisez "git restore --staged <fichier>..." pour désindexer)
    modifié :      Vagrantfile

Modifications qui ne seront pas validées :
  (utilisez "git add <fichier>..." pour mettre à jour ce qui sera validé)
  (utilisez "git restore <fichier>..." pour annuler les modifications dans le répertoire de travail)
    modifié :      Vagrantfile
```

3.3 Exercice 3 :

- Intégration des nouvelles modifications

– `$ git merge branche_1`

```
amadou@amadou:~/Bureau/IUT/Outils_Libres_Cote_client/git$ git merge
branche1  HEAD      master
amadou@amadou:~/Bureau/IUT/Outils_Libres_Cote_client/git$ git merge branche1
 Mise à jour 7927485..66e8410
Fast-forward
 Vagrantfile | 2 ++
1 file changed, 2 insertions(+)
```

- Oui la branche existe toujours, et pour la supprimer il faut :

– `$ git branch -d branche_1`

```
amadou@amadou:~/Bureau/IUT/Outils_Libres_Cote_client/git$ git branch
branche1
* master
amadou@amadou:~/Bureau/IUT/Outils_Libres_Cote_client/git$ git branch -d branche1
Branche branche1 supprimée (précédemment 66e8410).
amadou@amadou:~/Bureau/IUT/Outils_Libres_Cote_client/git$ git branch
* master
amadou@amadou:~/Bureau/IUT/Outils_Libres_Cote_client/git$
```

3.4 Exercice 3 bis :

- Création d'une nouvelle branche :

– `$ git checkout -b forwarder-new-port`

```
amadou@amadou:~/Bureau/IUT/Outils_Libres_Cote_client/git$ git checkout -b forwarder-new-port
Basculement sur la nouvelle branche 'forwarder-new-port'
amadou@amadou:~/Bureau/IUT/Outils_Libres_Cote_client/git$ git branch
* forwarder-new-port
  master
```

- Modification et commit :

```
config.vm.define "srv" do |b|
  b.vm.box = "debian/contrib-stretch64"
  b.vm.network "private_network", ip: "192.168.56.3"
  b.vm.network "forwarded_port", guest: 80, host: 8081
  b.vm.hostname = "srv"
```

- `$ git commit -a -m "forwarded-new-port"`

```
amadou@amadou:~/Bureau/IUT/Outils_Libres_Cote_client/git$ git status
Sur la branche forward-new-port
Modifications qui ne seront pas validées :
  (utilisez "git add <fichier>..." pour mettre à jour ce qui sera validé)
  (utilisez "git restore <fichier>..." pour annuler les modifications dans le répertoire de travail)
       modifié :       Vagrantfile

aucune modification n'a été ajoutée à la validation (utilisez "git add" ou "git commit -a")
amadou@amadou:~/Bureau/IUT/Outils_Libres_Cote_client/git$ git commit -a -m "forwarded-new-port"
[forward-new-port a565ba2] forwarded-new-port
1 file changed, 1 insertion(+)
```

- Modification sur le master avec le port 8080:

- `$ git checkout master`
- `$ git commit -a -m "port 8080"`

```
amadou@amadou:~/Bureau/IUT/Outils_Libres_Cote_client/git$ git checkout master
Basculement sur la branche 'master'
amadou@amadou:~/Bureau/IUT/Outils_Libres_Cote_client/git$ nano Vagrantfile
amadou@amadou:~/Bureau/IUT/Outils_Libres_Cote_client/git$ git commit -a
Abandon de la validation dû à un message de validation vide.
amadou@amadou:~/Bureau/IUT/Outils_Libres_Cote_client/git$ git commit -a -m "port 8080"
[master dc58a73] port 8080
1 file changed, 1 insertion(+)
```

- – Essaie de merge forward-new-port dans le master :

- `$ git merge forward-new-port`