Table des matières

TrinityCore pour les nuls ... 3
Installation ... 3
Pré-requis ... 3
Trinity ... 4
Checkout ... 4
Application de patches ... 4
Compilation ... 4
Extraction des maps ... 4
SQL ... 5
Bases world, characters et realmd ... 5
Traductions ... 6
Utilisation ... 6
Fichiers de conf ... 6
Démarrage ... 7
Firewall et NAT ... 8
Commandes, exemples, etc ... 8
Les bases ... 8
Tips ... 9
Liens ... 9
Troubleshooting ... 10

Doc ultra vieille et plus à jour
Allez plutôt ici : https://trinitycore.info/

TrinityCore pour les nuls

On compile from scratch un serveur WoW avec Trinity Core sous Debian. On peut aussi compiler sous Windows (il faut installer Visual C + +). Par ailleurs on doit pouvoir cross-compiler avec mingw mais je n'ai pas essayé.

⇒ Trinity est un fork de Mangos.

- Il n'y pas d'installeur en 1 clic
- Trinity / Mangos n'est pas facile à installer ni à utiliser, il faut être patient et lire les docs !
- Ca risque de ne pas fonctionner comme vous voulez du premier coup
- Les infos de cette page périment très vite vu que le projet évolue très vite (parfois + de 40 révisions par jour)
- Cette doc ne couvre pas MySQL, il faut donc avoir déjà des notions
- Monter un serveur *public* est illégal, un serveur *privé* est toléré semble t-il.

Vous aurez besoin :

- de temps
- de motivation
- de lire des docs
- de fouiller sur les forums
- de causer aux devs sur IRC

Vous obtiendrez :

- la satisfaction d'avoir monté un serveur de jeu fonctionnel à 90% avec les toutes dernières versions de WoW
- des compétences sous Linux & MySQL
- des compétences en dev (compilation, patches and co)

⇒ Version du client supportée : **3.1.3**
⇒ Version de la base de données : **TrinityDatabase 0.0.5**
⇒ Dernière maj de cette doc : **30/08/2009**
⇒ Versions utilisées pour cette doc : **5479** avec **TrinityDatabase 0.0.5**

- **Configuration matérielle** :
  - machine virtuelle Xen *{'cap': 0, 'weight': 512}*
  - 1 vCPU : 1 Xeon X3360 @ 2.8 Ghz
  - 3 Go de RAM
  - 30 Go d'espace disque
  - Bande passante : 1 Gb
- **Configuration logicielle** :
  - OS : Debian Lenny
  - Kernel : 2.6.26-2-xen-amd64
  - MySQL : 5.0.51a-24+lenny1
- **Statut du serveur** :
- World
- Realm

Installation

Pré-requis

Il faut parfois adapter, pour ma part j'ai installé sur une machine virtuelle Xen et certains packages diffèrent. Un *apt-get install build-essential* peut installer aussi certains packages du type *gcc* et compagnie. Certains packages existent peut-être déjà sur la machine, à vous de voir.

```
apt-get install zlibc gcc g++ make automake autoconf mysql binutils cpp libc6 libc6-dev libc6-i686 libgcc1 \
libmysql++-dev linux-header-`uname -r` linux-kernel-headers linux-kernel-devel makedepend mysql-client-5.0 \
mysql-common mysql-server python libssl0.9.8 libssl-dev openssl ssl-cert subversion g77 zlib1g-dev screen mc \
bunzip libstdc++ libtool unrar mercurial mercurial-common cmake
```

Il faut également installer la *libace*. Sous Debian Lenny on a la version 5.6.3 qui ne permettra pas de compiler le moteur. Il faut compiler une version plus récente à la mano en récupérant les sources ici. Choisir *Latest Release* et *ACE only*. Je suppose que vous savez compiler à partir des sources (configure, make, make install).

On créé un user pour ne pas lancer le truc en root :

```
adduser -d /home/wowsvr wowsvr
passwd wowsvr
cd /home/wowsvr
mkdir -p trinity2/work
cd trinity2/work
```

trinity**2** car c'est cette version qui fait fonctionner WOTLK, trinity**1** supporte jusqu'à la version 2.4.3 (Burning Crusade).

Trinity

Checkout

⇒ On utilise Mercurial pour récupérer les sources. C'est un peu comme un CVS ou un GIT. Plein de docs existent sur le net, donc je fais pas de grand discours la-dessus.

On récupère la dernière version. Ici on utilise directement la *patch queue*, c'est-à-dire qu'on récupère le code source original + des patches qu'on activera par la suite :

```
hg qclone http://bitbucket.org/panaut0lordv/tc2-patches/
cd tc2-patches
```

Application de patches

Les patches permettent de rajouter des fonctionnalités au moteur. Ci-dessous ceux que j'utilise :

Bots : Un patch qui vous permet d'embaucher un bot pour vous donner un coup de main (attaque, buffs, soins, etc) lors des phases d'XP → (aide), il permet aussi d'invoquer et de contrôler les persos de son compte (aide).
speedpatch : Augmente la vitesse nominale de deplacement des joueurs.
TeleNPC2 : Ajouter un bot qui permet de se téléporter pour optimiser les déplacements.

On sélectionne les patches et on les applique :

```
hg qsel bots_maximius_313 speedpatch TeleNPC2
hg qpush -a
```

Compilation

On configure le truc :

```
mkdir build
cd build
cmake ../ -DLARGE_CELL=1 -DDO_RA=1 -DPREFIX=/home/wowsvr/trinity2 -DCONF_DIR=/home/wowsvr/trinity2/etc
chown -R wowsvr:wowsvr /home/wowsvr/trinity2
```

Et on compile :

```
make
make install
```

il faut environ 30 min sur une VM Xen avec un proc à 2 Ghz et 15 min avec un Xeon à 2.8 GHz.

Extraction des maps

Ensuite on a besoin d'extraire les maps/vmaps du jeu ainsi que les *DBCs* qu'on doit placer sur le serveur. Ils sont indispensables pour le fonctionnement du serveur.

- DBCs et maps : */home/wowsvr/trinity2/work/tc2-patches/contrib/extractor/ad.exe*, à récupérer et à lancer sur le pc client pour extraire ce qu'il faut ;
- vmaps : */home/wowsvr/trinity2/work/tc2-patches/contrib/vmap_extract_assembler_bin/*, de la même façon que précédemment.

Une fois tout ces fichiers extraits, on les pose sur le serveur. Au final on a ça :

```
wowsvr@WoWvm:~/trinity2$ ls -ltr
total 288
drwxr-x--- 2 wowsvr wowsvr 114688 Aug 22 12:58 maps
drwxr-x--- 2 wowsvr wowsvr  12288 Aug 22 13:04 dbc
drwxr-x--- 2 wowsvr wowsvr   4096 Aug 22 14:49 lib
drwxr-x--- 2 root   root     4096 Aug 23 12:18 lost+found
drwxr-x--- 2 wowsvr wowsvr   4096 Aug 23 19:19 bin
drwxr-x--- 2 wowsvr wowsvr   4096 Aug 24 08:20 etc
drwxr-x--- 2 wowsvr wowsvr 135168 Aug 24 10:12 vmaps
```

SQL

Par défaut il n'y a pas de password root pour MySQL. C'est pas mal de mettre un password avant de commencer. La méthode facile et dégueu qui fonctionne est de tout faire en root. Pour ma part je créé toujours un user spécifique avec seulement les droits nécessaires. Ici un user *trinity* qui aura les droits sur les bases du jeu. Par la suite il faudra lancer des backups réguliers de base, etc. Je considère que vous savez déjà faire tout ça donc je ne m'étends pas ici.

Bases world, characters et realmd

On récupère la base de données UDB ainsi que la base TDB (on a juste besoin de quelques fichiers pour cette dernière) :

```
cd /home/wowsvr/trinity2/work
svn co http://svn.assembla.com/svn/trinitydatabase
```

On créé les bases de données, à vide, qui sont au nombre de 3 :

- **world** : la base qui contient le monde, elle peut-être supprimée et recrée si besoin ;
- **characters** : la base qui contient toutes les infos sur les persos, elle doit être manipulée avec précautions ;
- **realmd** : la base qui contient les infos du/des royaume(s) et des comptes, elle doit être manipulée avec précautions.

```
GRANT USAGE ON * . * TO 'trinity'@'localhost' IDENTIFIED BY 'somepassword' \
    WITH MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0 ;
CREATE DATABASE `world` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
CREATE DATABASE `characters` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
CREATE DATABASE `realmd` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
GRANT ALL PRIVILEGES ON `world` . * TO 'trinity'@'localhost' WITH GRANT OPTION;
GRANT ALL PRIVILEGES ON `characters` . * TO 'trinity'@'localhost' WITH GRANT OPTION;
GRANT ALL PRIVILEGES ON `realmd` . * TO 'trinity'@'localhost' WITH GRANT OPTION;
```

Puis on ajoute le contenu :

```
mysql -u trinity -p realmd < tc2-patches/sql/realmd.sql
mysql -u trinity -p characters < tc2-patches/sql/characters.sql

cd trinitydatabase/trunk/full_db/
unzip TDB_0.0.5_5254.zip
mysql -u trinity -p world < TDB_0.0.5_5254.sql

On met à jour les bases en passant les patchs Trinity dispos, on les passe **dans l'ordre croissant**, on démarre de 5254 jusqu'à la fin (ici 5314 pour la base world) et ce pour chaque base :

  * **world**

<code>
cd /home/wowsvr/trinity2/share/trinity/sql/updates/
ls -l *world* |tail -16
wowsvr@wowVM:~/trinity2/share/trinity/sql/updates$ ls -l *world* |tail -15
-rw-r--r-- 1 wowsvr wowsvr    77 Aug 23 10:39 5230_world_TDB.sql
-rw-r--r-- 1 wowsvr wowsvr    90 Aug 23 10:39 5230_world_scripts.sql
-rw-r--r-- 1 wowsvr wowsvr   471 Aug 23 10:39 5234_world_scripts.sql
-rw-r--r-- 1 wowsvr wowsvr   559 Aug 23 10:39 5237_world_scripts.sql
-rw-r--r-- 1 wowsvr wowsvr    88 Aug 23 10:39 5246_world_scripts.sql
-rw-r--r-- 1 wowsvr wowsvr    77 Aug 23 10:39 5254_world_scripts.sql
-rw-r--r-- 1 wowsvr wowsvr  1359 Aug 23 10:39 5270_world_scripts.sql
-rw-r--r-- 1 wowsvr wowsvr   114 Aug 23 10:39 5300_world_scripts_dk.sql
-rw-r--r-- 1 wowsvr wowsvr    84 Aug 23 10:39 5305_world_scripts.sql
-rw-r--r-- 1 wowsvr wowsvr  1346 Aug 23 10:39 5314_world_scripts_ulduar.sql
```

⇒ on démarre donc par le fichier *5270_world.sql*.

- **characters** et **realmd**

⇒ même principe

Puis on applique les derniers SQLs :

```
cd /home/wowsvr/trinity2/share/trinity/sql/FULL/
mysql -u trinity -p world < world_script_texts.sql
mysql -u trinity -p world < world_script_waypoints.sql
mysql -u trinity -p world < world_scripts_full.sql
mysql -u trinity -p world < world_spell_full.sql
```

A ce stade c'est terminé. Si vous avez utilisé les patches *Bots* et *TeleNPC2* il reste quelques petites manips à faire :

- **Bots** : se postionner sous */home/wowsvr/trinity2/share/trinity/sql/Bots*

```
world_bots.sql         -> à appliquer sur la base world
characters_bots.sql    -> à appliquer sur la base characters
```

- **TeleNPC2** : se postionner sous */home/wowsvr/trinity2/share/trinity/sql*

```
setup_npc_teleport.sql -> à appliquer sur la base world
```

les fichiers *world_bots.sql* et *setup_npc_teleport.sql* peuvent être customisés si besoin.

Traductions

On passe maintenant aux traductions :

```
cd /home/wowsvr/trinity2/work
svn co http://udbfr.googlecode.com/svn/
cd svn/trunk/Full_DB
unrar e UDBFR_0.0.6_MaNGOS_7685.rar
unrar e UDBFR_0.0.6_ScriptDev2_1012.rar
```

⇒ un coup de sed pour remplacer *mangos_string* par *trinity_string* dans les fichiers SQL

```
mysql -u trinity -p world < UDBFR_0.0.6_MaNGOS_7685.sql
mysql -u trinity -p world <  UDBFR_0.0.6_ScriptDev2_1012.sql
```

On choisit un nom pour le serveur, on ajoute son adresse IP (locale ou internet au choix) et on spécifie un password pour le user *trinity* :

```
mysql -u root -p
mysql> use realmd ;
mysql> UPDATE realmlist SET `name` = 'nom_du_serveur';
mysql> UPDATE realmlist SET `address` = 'AAA.BBB.CCC.DDD';
mysql> SET PASSWORD FOR trinity@localhost=PASSWORD('PASSWORD');
mysql> exit
```

Utilisation

Fichiers de conf

Dans les fichiers de conf suivants modifier le password ou les autres infos le cas échéant :

```
wowsvr@WoWvm:~/trinity2/etc$ grep 127.0.0.1 *
trinitycore.conf:LoginDatabaseInfo     = "127.0.0.1;3306;trinity;PASSWORD;realmd"
trinitycore.conf:WorldDatabaseInfo     = "127.0.0.1;3306;trinity;PASSWORD;trinity"
trinitycore.conf:CharacterDatabaseInfo = "127.0.0.1;3306;trinity;PASSWORD;characters"
trinityrealmd.conf:LoginDatabaseInfo = "127.0.0.1;3306;trinity;PASSWORD;realmd"
```

Modifier aussi la directive *DataDir* dans *trinitycore.conf* pour pointer vers le répertoire qui contient les répertoires *vmaps*, *maps* et *dbc*.

Le paramètre *vmap.enableLOS = 1* permet une meilleure gestion de l'espace.

Le paramètre *LogsDir* permet d'éviter d'avoir les logs dans le répertoire *bin*.

On peut aussi modifier le fichier de conf *trinitycore.conf* pour changer le jeu :

```
wowsvr@WoWvm:~/trinity2/etc$ grep -i Rate trinitycore.conf |egrep "XP|Money"|egrep -v "^#|70"
Rate.Drop.Money = 5
Rate.XP.Kill    = 5
Rate.XP.Quest   = 5
Rate.XP.Explore = 5
```

La on multiplie l'XP par 5 lors d'un kill de mob, d'une qûete ou de la visite d'une nouvelle zone. L'argent est aussi multiplié par 5. Le fichier est bien commenté et contient pas mal de paramètres.

Le fichier est bien documenté.

Bravo, vous avez presque terminé ;)

Démarrage

Pour démarrer, on peut créer un script */home/wowsvr/trinity2/bin/trinity.sh* :

```
#!/bin/bash

[[ $# -ne 1 ]] && echo "./trinity.sh stop ou ./trinity.sh start" && exit

case $1 in

        start)

        for i in realmd worldd
        do
        screen -A -m -d -S $i ~/trinity2/bin/trinity-$i
        echo "Lancement de trinity-$i : OK"
        done
        ;;

        stop)

        for i in realmd worldd
        do
        killall trinity-$i
        echo "Arret de trinity-$i : OK"
        done
        ;;

        restart)

        ~/trinity2/bin/trinity.sh stop
        echo
        ~/trinity2/bin/trinity.sh start

        ;;

        status)
        clear
        for i in realmd worldd
        do
        echo "Status de trinity-$i"
        lsof -a -p `ps au|grep $i|grep -v grep|awk '{print $2}'` -i 2>/dev/null|grep LISTEN
        echo
```

```
        done

        ;;

        *)

        echo "Mauvais choix ... stop, start, restart ou status"
esac
```

Puis on lance le tout :

```
cd ~/trinity2/bin
./trinity.sh start
```

Pour lister les screens :

```
screen -list
```

Pour s'y connecter :

```
screen -r nom_du_screen
```

Firewall et NAT

Pour un serveur online il faut forwarder les ports TCP 3724 et 8085 de votre routeur vers la machine qui héberge le serveur. En parallèle je n'autorise que certaines IPs :

```
for port in 3724 8085
do
/sbin/iptables -A INPUT -p tcp --dport $port -j DROP
done

for ip in AAA.BBB.CCC.DDD EEE.FFF.GGG.HHH
do
for port in 3724 8085
do
/sbin/iptables -I INPUT -p tcp --sport 1024: --dport $port -s $ip -m state --state NEW,ESTABLISHED -j ACCEPT
/sbin/iptables -I OUTPUT -p tcp --dport 1024: --sport $port -s $ip -m state --state ESTABLISHED -j ACCEPT
done
done
```

Commandes, exemples, etc

Les bases

Si tout est OK ca donne ca :

```
wowsvr@WoWvmII:~$ screen -r
There are several suitable screens on:
        25579.trinity-core      (06/24/09 20:30:35)     (Detached)
        25576.trinity-realm     (06/24/09 20:30:35)     (Detached)
Type "screen [-d] -r [pid.]tty.host" to resume one of them.

wowsvr@WoWvm:~$ screen -r 25579.trinity-core
TC>help
TC>Syntax: .help [$command]

Display usage instructions for the given $command. If no $command provided show list available commands.
Commandes disponibles:

    account ...
    gm ...
    tele ...
    reload ...
    list ...
    lookup ...
    pdump ...
    guild ...
```

```
    reset ...
    instance ...
    server ...
    ahbotoptions
    announce
    gmannounce
    notify
    gmnotify
    commands
    revive
    help
    saveall
    kick
    ban ...
    unban ...
    banlist ...
    pinfo
    send ...
    loadscripts
    mute
    unmute
    repairitems
TC>
```

- Créer un compte GM avec WOTLK activé :

```
TC> account create toto password
TC> account set addon 2
TC> accoutn set gmlevel 3
```

Tips

- Une fois GM on peut utiliser tout un tas de commandes :

.level : pour passer un level
.level 10 : prendre 10 levels
.modify money 100000 : ajouter 100 000 pièces de cuivre

D'autres commandes ici.

On peut créer des armes en bidouillant la base de données (cf. http://wiki.udbforums.org) et même certains sites mâchent le travail : http://wow-v.com. On créé son arme et le site génère le code SQL à injecter dans la base de données. On peut faire la même chose pour les mobs, les PNJs, etc. On peut aussi s'attribuer des objets, par ex. celui-ci, il suffit de taper *.additem 34540*, ou bien chercher les compétences dans la base de données et se les attribuer avec *.learn* ou *.setskill*.

- Trouver les ID :

Soit en ouvrant la base de données et en regardant dans les tables : *item_template* pour les items, *creature_template* pour les creatures. Pour les spells , les locations des .tele , etc (regardez les contenus des tables tout est là).

Soit en allant sur ces sites pour récupérer les IDs dans l'URL :

http://wow.allakhazam.com
http://www.thottbot.com
http://www.wowdbu.com
http://worldofwarcraft.judgehype.com
http://www.wowhead.com
http://wow.mmodb.com

MiniManager est un frontend web très utile pour gérer ses serveurs, cf. plus bas.

Liens

- Monter son serveur sous Linux : http://www.crashdump.fr/world-of-warcraft/creer-son-serveur-world-of-warcraft-prive-avec-mangos-mangoslinux-52/
- Les commandes GM : http://www.gameguides.ro/wow-trinity-core-gm-commands.html
- Une base de données WoW : http://wiki.udbforums.org
- La base de données pour Trinity : http://www.trinitydatabase.org
- Quice, un éditeur de base de données : http://quice.indomit.ru
- MiniManager : https://mmfpm.svn.sourceforge.net/svnroot/mmfpm
- Dumper son perso d'un serveur officiel : http://www.manground.org/charactercopy.html ne fonctionne plus
- Améliorer la qualité graphique : http://wowdb.fr/showthread.php?tid=8474&page=1
- Bitbucket TrinityCore 2 : http://dev.trinitycore.org/trinitycore2
- Patchqueue TrinityCore 2 : http://bitbucket.org/panaut0lordv/tc2-patches

- Fun SQLs : http://udbforums.org/index.php?topic=2819.0
- Pour se simplifier la vie, un addon GM pour WoW : http://code.google.com/p/thegm/downloads/list
- Convertisseur : http://bitbucket.org/Drahy/trinity-converters/

Troubleshooting

- Erreur *Unit 61 (Type: 4) has invalid combat_reach 0.000000*

→ 61 est le GUID

Sélectionner le perso (en tant que GM) et taper *.debug getvalue 151* qui doit être égal à 0. Récupérer la valeur sur un perso de la même race/sexe. Et faire un *.debug setvalue 151 #le_resultat* sur le perso buggé pour mettre la bonne valeur.

- Faire du ménage dans la table *item_instance*

```
select * from `item_instance` where `owner_guid` = <AHbot Account ID> and `guid` not in (select `itemguid` from `auctionhouse` UNION select `item` from `character_inventory` UNION select `item_guid` from `guild_bank_item` UNION select `item_guid`
from `mail_items` UNION select `item_guid` from `character_gifts`);
```

From:
https://unix.ndlp.info/ - **Where there is a shell, there is a way**

Permanent link:
https://unix.ndlp.info/doku.php/informatique:jeux_videos:jeux_actuels_et_ou_recents:world_of_warcraft:serv3.0.9:trinity

Last update: **2025/08/27 10:58**