

Cheat Sheet indispensable en DUT INFO S3

feuille publiée par erreur mais en cours d'amélioration :-)

Pour qui ?

les DUT INFO S3 de Nice - Fabron

Dans quel module ?

M311 / principes des systèmes d'exploitations

Pourquoi ?

être dix fois plus efficace en TP

Ouvrir un terminal GNOME

Depuis le bureau GNOME : Cliquer en haut à gauche sur "Activités" puis **Q** Rechercher "terminal" dans la zone de recherche en haut du bureau. En profiter pour placer l'icône dans la barre "Activités"

Depuis un terminal GNOME : Menu "Fichier" puis "Ouvrir un terminal"

Depuis l'explorateur de fichiers : Clic droit dans le fond de la fenêtre et "Ouvrir dans un terminal"

Le clic droit dans l'explorateur permet d'ouvrir le terminal dans le bon répertoire

Ne pas hésiter à ouvrir plusieurs terminaux côte à côte pour tester les programmes multi processus et contrôler leur exécution depuis un autre terminal (avec `kill`, `ps`, `top`, `htop`, ..)

Taper des commandes dans le terminal

Copier / Coller : surligner le texte choisi à la souris puis clic gauche dans le terminal et clic milieu pour coller

Complétion : `tab` et/ou `tab` pour faire défiler les possibilités en boucle puis taper un ou plusieurs caractères pour choisir

Historique bash : taper la commande `history`

Commandes précédentes : **↑** ou `ctrl-p`

Commandes suivantes : **↓** ou `ctrl-n`

Début de ligne : `ctrl-home` ou `ctrl-a`

Taper des commandes dans le terminal (cont)

Fin de ligne : `ctrl-end` ou `ctrl-e`

Abandonner : `ctrl-c`

Rechercher la dernière commande contenant `mot` : `ctrl-r` `mot` puis `ctrl-r` autant de fois que nécessaire pour retrouver toutes les commandes contenant `mot`

Couper jusqu'à la fin de ligne : `ctrl-k`

Coller le texte coupé : `ctrl-y`

Pour les autres raccourcis, voir la documentation du mode emacs

Voir ses fichiers Windows depuis Linux

Dans l'explorateur de fichiers, taper dans la barre du haut (ou `ctrl-i`) et entrer la connexion `smb://dc-info-03/users/etudiant/login`
domaine : IUTNICE
password : celui utilisé sous Windows
Les supports de cours des autres modules sont sur le partage :
`smb://fs-info-01/SupportCours/S3T`

Commandes bash de base

Afficher le contenu du répertoire `dir` : `ls -la dir`

Se positionner à la racine de son espace personnel : `cd` ou `cd ~` ou `cd $HOME`

`$HOME` et `~` désignent le même chemin, le répertoire défini comme racine du login correspondant, par exemple `/home/martin`

Effacer le répertoire `dir` et ses sous-répertoires : `rm -rf dir`

Renommer `dir1` sous le nom de `dir2` : `mv dir1 dir2`

Effacer le terminal : `clear`

Ajouter `$HOME/bin` au `PATH` : `export PATH=$PATH:$HOME/bin`

Changer l'invite de commande : `export PS1='$ '`

Les variables d'environnement comme `PS1`, `PATH`, ... sont à changer de préférence dans les fichiers de configuration de bash : `.bash_login`, `.bash_profile`, `.bashrc`, ...

Gestion zéro des archives tar.gz

Extraire le contenu de `tp.tar.gz` `tar xzvf tp.tar.gz`

Créer une archive contenant le répertoire `tp` `tar zcvf tp.tar.gz tp`

Pour créer l'archive contenant tous les fichiers du répertoire `tp`, il faut se positionner avant au dessus de `tp`. Par exemple, si le chemin absolu de `tp` est `/home/login/ASR/tp` alors les commandes sont:

```
cd /home/login/ASR
tar zcvf tp.tar.gz tp
```

Commandes de contrôle des processus

`ps axu` ou bien `ps -eF` voir tous les processus en cours

`ps axjf` ou bien `ps -ejH` voir aussi les arbres de filiation

`pstree` affiche un arbre de tous les processus en cours

`ps -elfT` ou bien `ps axms` avoir en plus les informations sur les threads

`top` ou bien `top U login` affichage dynamique, `top -H` pour voir les threads

`htop` ou bien `htop -u login` taper `H` pour voir les threads, `t` pour les arbres, `F4` pour filtrer un nom

`kill -SIGINT pid` interrompre le processus numéro `pid`

`kill -SIGKILL pid` tuer le processus numéro `pid` qui résiste à `SIGINT`

`killall -INT prog` interrompre tous les exemplaires de processus de nom `prog`

`killall -KILL prog` tuer tous les exemplaires des processus de nom `prog`

Éditer, compiler et exécuter des programmes C

Éditer avec `nano`, `gedit`, `notepad++`, `emacs`, `vim`

ou un autre éditeur à partir du moment que le codage est UTF-8

Compiler `gcc -Wall -std=gnu99 prog.c -o prog`

Compiler avec un Makefile `make`

Nettoyer le répertoire courant `make mrproper` ou `make clean`

Exécuter `prog` `./prog`

Éditer, compiler et exécuter des programmes C (cont)

Interrompre le processus `prog` `ctrl-c` ou `ctrl-\`

Formater un source C `indent -linux prog.c`

Réutiliser systématiquement les exemples de Makefile et de sources C donnés avec le sujet de TD

Exemple

```
$ export CC=gcc
$ cd ~/ASR/M211_TP/TP_6_IPC/tp3
$ ls
consommateur.c  msg_max_default_10.txt  produit_scalaire_k  puissance.c
consommateur.c  producteur.c             produit_scalaire_k.c  somme
Makefile        producteur.c             puissance
$ make mrproper
$ make
gcc -xextra -Wall -g -std=gnu99 -pthread -lrt -lm puissance.c -o puissance
gcc -xextra -Wall -g -std=gnu99 -pthread -lrt -lm somme.c -o somme
gcc -xextra -Wall -g -std=gnu99 -pthread -lrt -lm producteur.c -o producteur
gcc -xextra -Wall -g -std=gnu99 -pthread -lrt -lm consommateur.c -o consommateur
gcc -xextra -Wall -g -std=gnu99 -pthread -lrt -lm produit_scalaire_k.c -o produit_scalaire_k
$ ./somme 2 10
Résultat : 2047
$
```

sur cet exemple (PuTTY) on montre comment

- 1 - changer l'invite de commande
- 2 - se positionner dans un répertoire depuis son `$HOME` (ou `~`)
- 3 - effacer les exécutables avec la règle `mrproper` du Makefile pour
- 4 - tout recompiler et
- 5 - exécuter le programme `somme`

Erreur de compilation bizarre

error stray '204' in program ... effacer puis retaper le texte

Erreur souvent due à un copier/coller du web vers gedit : il faut télécharger les sources, pas faire des copier/coller

Trouver les #include<>

Pour chaque appel système ou routine de la libc, consulter le manuel

Exemple pour `fork()` : taper

`man fork`

qui nous indique d'ajouter

`#include <unistd.h>`

Dans le cas où il existe des entrées multiples dans le manuel, alors il faut préciser la section. Exemple :

`man 3 sleep` affiche la page de manuel de la fonction `Csleep()`

`man sleep` affiche la page de manuel de la commande `bashsleep` qui est trouvée en premier car elle est dans la section 1 (et non 3) du manuel

Pour trouver toutes les entrées (et les sections correspondantes) contenant mot : `man k mot`