



Informatique

Travaux Pratiques

Axel LE BOT



Copyright © 2017-2018 Axel LE BOT

Licensed under the Creative Commons Attribution-NonCommercial 3.0 Unported License (the “License”). You may not use this file except in compliance with the License. You may obtain a copy of the License at <http://creativecommons.org/licenses/by-nc/3.0>. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.



Contents

I

Shell

1	TP1 : Manipulations de l'environnement et des fichiers sous UNIX .	9
1.1	Exercice 1 : Découverte de quelques commandes d'archivage	9
1.1.1	1. Récupération et décompression d'une archive	9
1.1.2	2. Manipulation de fichiers	9
1.1.3	3. Création d'une nouvelle archive	9
1.2	Exercice 2 : Utilisation des masques de création de fichiers	10
1.2.1	1.	10
1.2.2	2. et 3.	10
1.3	Exercice 3 : Manipulation du Systeme de fichier et des droits de navigation	10
1.3.1	1.	10
1.3.2	3.	10
1.3.3	4.	10
1.3.4	6.	10
1.3.5	7.	10
1.4	Exercice 4 : Manipulation d'expression régulière	10
1.4.1	2.	10
1.4.2	3.	10
1.4.3	4.	11
1.4.4	5.	11
1.4.5	6.	11
1.4.6	7.	11
1.4.7	8.	11
1.4.8	9.	11

1.4.9	10.	11
1.4.10	11.	11
1.4.11	12.	11
1.4.12	13.	11
1.4.13	15.	11
1.4.14	16.	11
1.4.15	17.	11

2	TP2 : Scripts Shell	13
2.1	Exercice 5 : Un premier script	13
2.2	Exercice 6 : Comptage des paramètres	13
2.3	Exercice 7 : Portée des variables	14

II

C++

3	TP1+TP2 : Tableaux, matrices et Fonctions récursives	17
3.1	Exercice sur des tableaux	17
3.1.1	Fonction sur les tableaux non triés	17
3.1.2	Algorithmes de tri de tableaux	17
3.1.3	Fonctions sur les tableaux triés	17
3.1.4	Exercice sur les Fonctions récursives	17
4	TP3+TP4 : Manipulation des arbres	19
4.1	Algorithmes sur arborescences binaires de recherche (ABR) non équilibrées	19
4.1.1	Exercice 1 : Mise en place d'ABR et premiers algorithmes	19
4.1.2	Exercice 2 : Algorithmes récursifs sur arborescences	19
4.2	Modification et parcours d'ABR non équilibrées	19
4.2.1	Exercice 3 : Insertion et suppression de valeurs dans une arborescence	19
4.3	Parcours d'arborescences binaires	19
4.3.1	Exercice 4 : Parcours sur arbres	19
5	TP5+TP6 : Hiérarchie de processus, signaux	21
5.1	Gestion des signaux : envoi et reception	21
5.1.1	Exercice 1 : Droits et signaux	21
5.1.2	Exercice 2 : capture de signal et traduction en langage C	21
5.1.3	Exercice 3 : Capture de signaux et redirections (exercice difficile)	21
5.1.4	Exercice 4 : envoi multiples et capture de signal en C	21
5.2	Gestion des processus	21
5.2.1	Exercice 5 : Processus en premier-plan / Arriere-plan	21
5.2.2	Exercice 6 : Duplication et recouvrement de processus	21
5.3	Gestion des processus - Suite	21
5.3.1	Exercice 7 : Duplication de processus	21
5.3.2	Exercice 8 : Creation et destruction de processus	21
5.3.3	Exercice 9 : Evaluation du nombre de processus	21
5.3.4	Exercice 10 : Conjonctions, Disjonctions, et Duplication	21

5.3.5	Exercice 11 : Terminaison normale de processus	21
-------	--	----

6 TP7+TP8 : Communication socket 23

6.1 Communication distante en utilisant l'outil netcat 23

6.1.1	Exercice 1 : Découverte de la commande nc : netcat	23
-------	--	----

6.1.2	Exercice 2 : Utilisation de la commande nc : netcat pour le transfert de fichier et l'évaluation de la bande passante	23
-------	---	----

6.1.3	Exercice 3 : Une histoire de serveurs concurrents	23
-------	--	----

6.1.4	Exercice 4 : Comprendre une requête HTTP	23
-------	--	----

6.2 Développement d'un client et d'un serveur en C 23

6.2.1	Exercice 5 : Mise en place d'une communication en mode non connecte ..	23
-------	--	----

6.2.2	Exercice 6 : Création d'une architecture (client UDP) - (relai UDP-TCP)- (serveur TCP)	23
-------	--	----

6.3 Exercices bonus 23

6.3.1	Exercice 7 : Résolution de noms	23
-------	---------------------------------------	----

6.3.2	Exercice 8 : Serveur multi-client en mode connecte	23
-------	--	----

7 TP9+TP10 : Héritage multiple et modélisation 25

7.1	Exercice 1 : Organisation d'un jeu de combat au tour par tour	25
-----	---	----

8 TP11+TP12 : Projet de synthèse 27

8.1	Realisation d'un jeu de Puissance 4	27
-----	-------------------------------------	----

III

Autres

8.2	Figure	31
-----	--------	----

	Bibliography	33
--	---------------------------	-----------

	Books	33
--	--------------	-----------

	Articles	33
--	-----------------	-----------

	Index	35
--	--------------------	-----------



Shell

1	TP1 : Manipulations de l'environnement et des fichiers sous UNIX	9
1.1	Exercice 1 : Découverte de quelques commandes d'archivage	
1.2	Exercice 2 : Utilisation des masques de création de fichiers	
1.3	Exercice 3 : Manipulation du Systeme de fichier et des droits de navigation	
1.4	Exercice 4 : Manipulation d'expression régulière	
2	TP2 : Scripts Shell	13
2.1	Exercice 5 : Un premier script	
2.2	Exercice 6 : Comptage des paramètres	
2.3	Exercice 7 : Portée des variables	

UNIX

bash# where there is a shell, there is a way■

1. TP1 : Manipulations de l'environnement et des fi

1.1 Exercice 1 : Découverte de quelques commandes d'archivage

L'objectif de cet exercice est de découvrir et manipuler les commandes de téléchargement, d'archivage, de compression et de décompression de fichier

1.1.1 1. Récupération et décompression d'une archive

La commande `wget https://cloud.infotro.fr/ITC313/archive.tar` permet de télécharger l'archive présent à cette adresse.

- L'option `-x` permet de restaurer les fichiers contenus dans une archive.
- L'option `-c` permet de créer une nouvelle archive.
- L'option `-f` permet d'utiliser le fichier archive F ou le périphérique F (par défaut `/dev/rmt0`).

9 fichiers étaient présents dans cette archive

1.1.2 2. Manipulation de fichiers

file `.*` Afin de renommer le fichier j'utilise cette commande `mv image4.jpg image4.jpg2`. Le fichier `script.txt` fait 170Ko. La commande `gzip` sur `script.txt` a compressé le fichier. Le fichier fait maintenant 65Ko. La compression est donc d'environ 38.235%. Après décompression avec la commande `gunzip` le fichier fait maintenant 170Ko, qui est la taille initiale du fichier.

1.1.3 3. Création d'une nouvelle archive

La nouvelle archive fait 850Ko soit 1Ko de moins que l'ancienne archive. Sûrement à cause du nom de fichier `jp2` changé. La somme des tailles des fichiers dans l'archive est égale à 845479 soit 845Ko, on observe une différence de 5Ko. La compression de l'archive (créée précédemment) fait 617Ko soit une différence de 228Ko. L'option `-z` utilise `gzip` pour compresser l'archive. Elle revient totalement à créer une archive puis la compresser puisque d'après les tests la taille n'est pas différente. La Commande `tar -c -z *.jpg *.txt *.jp2` devrait normalement afficher dans le terminal le résultat. La Commande `tar -c -z *.jpg *.txt *.jp2 > nouvelleArchive3.tar.gz` redirige bien le résultat dans un fichier. La redirection du flux dans un fichier recrée une archive

comprimé similaire à la deuxième créé. En conclusion l'archive 2 et 3 donne le même résultat et sont plus petit que l'archive 1 puisqu'elles sont comprimées.

1.2 Exercice 2 : Utilisation des masques de création de fichiers

1.2.1 1.

```
1          $ touch Raphael.txt
2          $ umask 0666
3          $ touch Donatello.txt
4          $ umask 0331
5          $ touch Michelangelo.txt
6          $ umask 0661
7          $ touch Leonardo.txt
8          $ umask 0000
```

1.2.2 2. et 3.

Il n'est pas possible de créer de donner plus de droit que la limitation par défaut du système. (application umask par défaut 666 sur fichier et 777 sur repertoire)

1.3 Exercice 3 : Manipulation du Système de fichier et des droits de navigation

1.3.1 1.

L'archive contient 5 images.

1.3.2 3.

```
/home/ESIREM-AD/al669724/Documents/Shell/TPs/TP1/Ex3/images/Chinpokomon/P-Z/Vamporc.png
```

1.3.3 4.

```
../P-Z/Vamporc.png
```

1.3.4 6.

L'option -z permet de compresser l'archive.

```
$ tar -xczf ITC313_TP_Shell_lebot.axel.tar.gz
```

permettra de décompresser et extraire l'archive.

1.3.5 7.

Toutes les permissions sont conservées.

1.4 Exercice 4 : Manipulation d'expression régulière

1.4.1 2.

Les lignes contenant la suite de lettres "ette".

1.4.2 3.

Les lignes contenant la lettre "T".

1.4.3 4.

Les lignes commençant par la lettre "T".

1.4.4 5.

^ signifie "début".

1.4.5 6.

Les lignes finissant par "te".

1.4.6 7.

Les lignes contenant la suite de caractère "c", un caractère, "r".

1.4.7 8.

Les lignes contenant "oui" ou "non".

1.4.8 9.

- '\$' -> en fin
- '|' -> ou
- '.' -> un caractère

1.4.9 10.

Permet d'afficher uniquement la partie correspondant à la recherche.

1.4.10 11.

Une suite de 4 lettre Majuscule.

1.4.11 12.

Une suite d'au moins une Majuscule et un minuscule

1.4.12 13.

Les mots commençant par une majuscule ainsi que les lettres majuscules.

1.4.13 15.

Permet de récupérer les adresses e-mail.

1.4.14 16.

Permet de récupérer les numéros de téléphone.

1.4.15 17.

((bien))((joue))((tu))((as))((trouve))((la))((reponse))((a))((la))((derniere))((question))

2. TP2 : Scripts Shell

2.1 Exercice 5 : Un premier script

```
1 #!/bin/bash
2 # Axel LE BOT - 2017-10-01
3 clear
4 USER=$(whoami)
5 echo "Hello $USER"
6 echo -e "\n\e[9mhello world\e[0m\n \"hello Kitty\""
7 echo "Do you like fishsticks(y/n)?"
8 read answer
9
10 if [ "$answer" = "y" ]
11 then
12     echo "then you are a gayfish!"
13 else
14     echo "OK"
15 fi
```

2.2 Exercice 6 : Comptage des paramètres

```
1 #!/bin/bash
2 # Axel LE BOT - 2017-10-02
3 echo "Ex6 - param tres"
4 echo -n "-Liste des parametres entres : "
5 COUNT=0
6
7 for i in $*
8 do
9     echo -n "$i "
10    (( COUNT++ ))
```

```
11 done
12
13 echo -e "\nNombre de parametres : $COUNT"

1 #!/bin/bash
2 # Axel LE BOT - 2017-10-02
3 echo "Ex6 - param tres"
4 echo -n "-Liste des parametres entres : "
5
6 A=$#
7 for ((i=0 ; i<A ; i++)) do
8     echo -n "$1 "
9     shift
10 done
11
12 echo -e "\nNombre de parametres : $A"
```

2.3 Exercice 7 : Portée des variables

1. Portée des variables locales

La variable créé dans le terminal n'est pas accessible depuis un script.

2. Portée limitée au shell

Les variables sont local au terminal

3. Étendre la portée de la valeur d'une variable locale

3	TP1+TP2 : Tableaux, matrices et Fonctions récursives	17
3.1	Exercice sur des tableaux	
4	TP3+TP4 : Manipulation des arbres	19
4.1	Algorithmes sur arborescences binaires de recherche (ABR) non équilibrées	
4.2	Modification et parcours d'ABR non équilibrées	
4.3	Parcours d'arborescences binaires	
5	TP5+TP6 : Hiérarchie de processus, signaux	21
5.1	Gestion des signaux : envoi et reception	
5.2	Gestion des processus	
5.3	Gestion des processus - Suite	
6	TP7+TP8 : Communication socket	23
6.1	Communication distante en utilisant l'outil netcat	
6.2	Développement d'un client et d'un serveur en C	
6.3	Exercices bonus	
7	TP9+TP10 : Héritage multiple et modélisation	25
7.1	Exercice 1 : Organisation d'un jeu de combat au tour par tour	
8	TP11+TP12 : Projet de synthèse	27
8.1	Realisation d'un jeu de Puissance 4	

3. TP1+TP2 : Tableaux, matrices et Fonctions récursives

3.1 Exercice sur des tableaux

3.1.1 Fonction sur les tableaux non triés

Exercice 1 : Algorithmes de parcours classiques sur tableau non triés

Quelques copier coller ont suffi. Les tests ont bien été effectués.

Exercice 2 : Ajout et suppression d'éléments tableaux non triés

3.1.2 Algorithmes de tri de tableaux

Exercice 3 : Trier des tableaux aléatoires

3.1.3 Fonctions sur les tableaux triés

Exercice 4 : Algorithmes de parcours classiques sur tableau non triés

Exercice 5 : Ajout et suppression d'éléments sur tableaux triés

3.1.4 Exercice sur les Fonctions récursives

Exercice 6 : Définition de fonction récursive

Exercice 7 : Algorithme récursif sur matrice



4. TP3+TP4 : Manipulation des arbres

- 4.1** Algorithmes sur arborescences binaires de recherche (ABR) non équilibrées
 - 4.1.1** Exercice 1 : Mise en place d'ABR et premiers algorithmes
 - 4.1.2** Exercice 2 : Algorithmes récursifs sur arborescences
- 4.2** Modification et parcours d'ABR non équilibrées
 - 4.2.1** Exercice 3 : Insertion et suppression de valeurs dans une arborescence
- 4.3** Parcours d'arborescences binaires
 - 4.3.1** Exercice 4 : Parcours sur arbres


```

top - 20:45:35 up 620, 1 user, load average: 3.99, 3.57, 2.90
Tasks: 103 total, 2 running, 101 sleeping, 0 stopped, 0 zombie
Cpu(s): 62.9%us, 31.8%sy, 0.0%ni, 0.0%id, 0.0%wa, 0.3%hi, 5.0%si, 0.0%st
Mem: 1020296k total, 954376k used, 65920k free, 8704k buffers
Swap: 2064376k total, 206728k used, 1857648k free, 221048k cached

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
8904	root	20	0	103m	1036	504	S	48.7	0.1	4:25.21	htpd
10401	root	20	0	552m	297m	2432	R	23.2	29.9	66:34.02	srcds_linux
9421	root	20	0	578m	311m	2508	S	21.9	31.3	61:49.56	srcds_linux
3000	root	20	0	31704	636	204	S	6.0	0.1	2:25.62	owjumioig
11533	root	20	0	15036	1220	936	R	0.3	0.1	0:00.02	top
1	root	20	0	19356	1152	940	S	0.0	0.1	0:04.73	init
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
4	root	20	0	0	0	0	S	0.0	0.0	3:40.87	ksoftirqd/0
5	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
6	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	watchdog/0
7	root	20	0	0	0	0	S	0.0	0.0	0:09.51	events/0
8	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cgroup

5. TP5+TP6 : Hiérarchie de processus, signaux

5.1 Gestion des signaux : envoi et reception

- 5.1.1 Exercice 1 : Droits et signaux
- 5.1.2 Exercice 2 : capture de signal et traduction en langage C
- 5.1.3 Exercice 3 : Capture de signaux et redirections (exercice difficile)
- 5.1.4 Exercice 4 : envoi multiples et capture de signal en C

5.2 Gestion des processus

- 5.2.1 Exercice 5 : Processus en premier-plan / Arriere-plan
- 5.2.2 Exercice 6 : Duplication et recouvrement de processus

5.3 Gestion des processus - Suite

- 5.3.1 Exercice 7 : Duplication de processus
- 5.3.2 Exercice 8 : Creation et destruction de processus
- 5.3.3 Exercice 9 : Evaluation du nombre de processus
- 5.3.4 Exercice 10 : Conjonctions, Disjonctions, et Duplication
- 5.3.5 Exercice 11 : Terminaison normale de processus



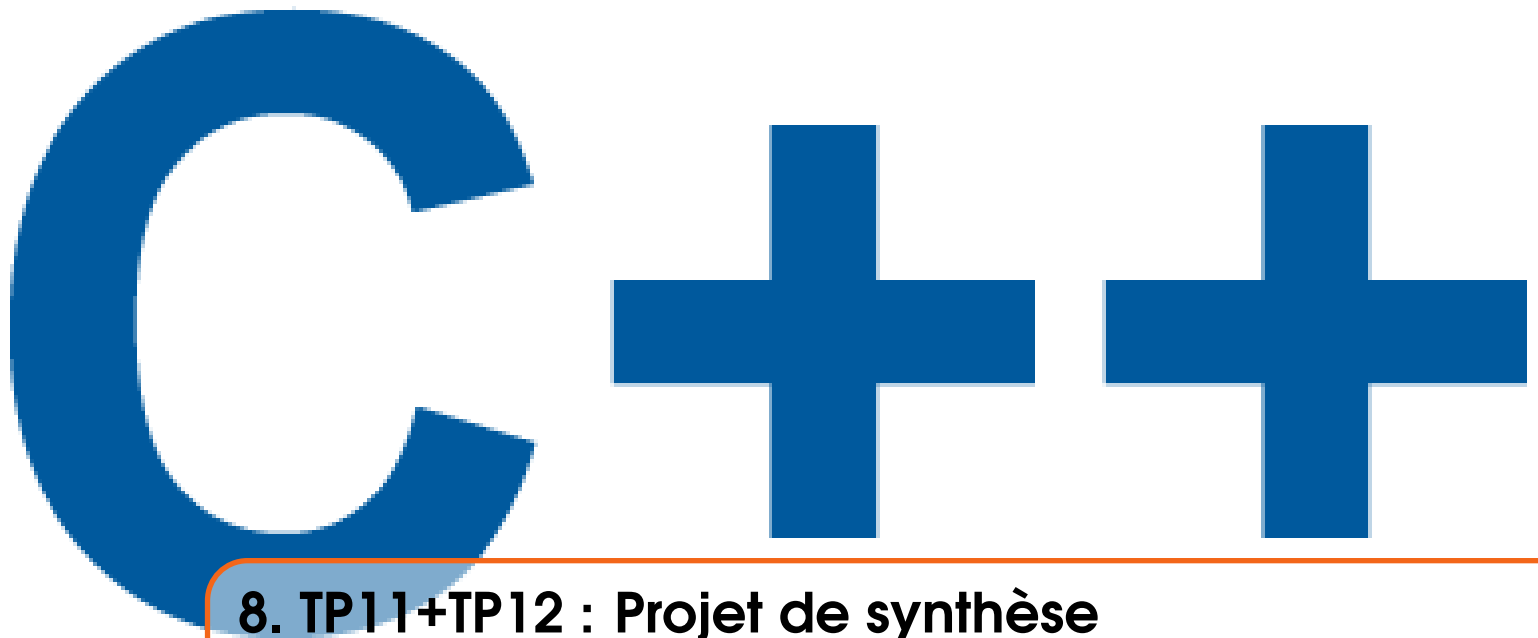
6. TP7+TP8 : Communication socket

- 6.1** Communication distante en utilisant l'outil netcat
 - 6.1.1** Exercice 1 : Découverte de la commande nc : netcat
 - 6.1.2** Exercice 2 : Utilisation de la commande nc : netcat pour le transfert de fichier et l'évaluation de la bande passante
 - 6.1.3** Exercice 3 : Une histoire de serveurs concurrents ...
 - 6.1.4** Exercice 4 : Comprendre une requête HTTP
- 6.2** Développement d'un client et d'un serveur en C
 - 6.2.1** Exercice 5 : Mise en place d'une communication en mode non connecte
 - 6.2.2** Exercice 6 : Création d'une architecture (client UDP) - (relai UDP-TCP)- (serveur TCP)
- 6.3** Exercices bonus
 - 6.3.1** Exercice 7 : Résolution de noms
 - 6.3.2** Exercice 8 : Serveur multi-client en mode connecte

C++

7. TP9+TP10 : Héritage multiple et modélisation

7.1 Exercice 1 : Organisation d'un jeu de combat au tour par tour



8. TP11+TP12 : Projet de synthèse

8.1 Realisation d'un jeu de Puissance 4



Autres

8.2 Figure

Bibliography 33
Books
Articles

Index 35

8.2 Figure

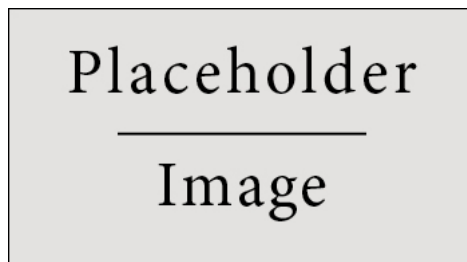
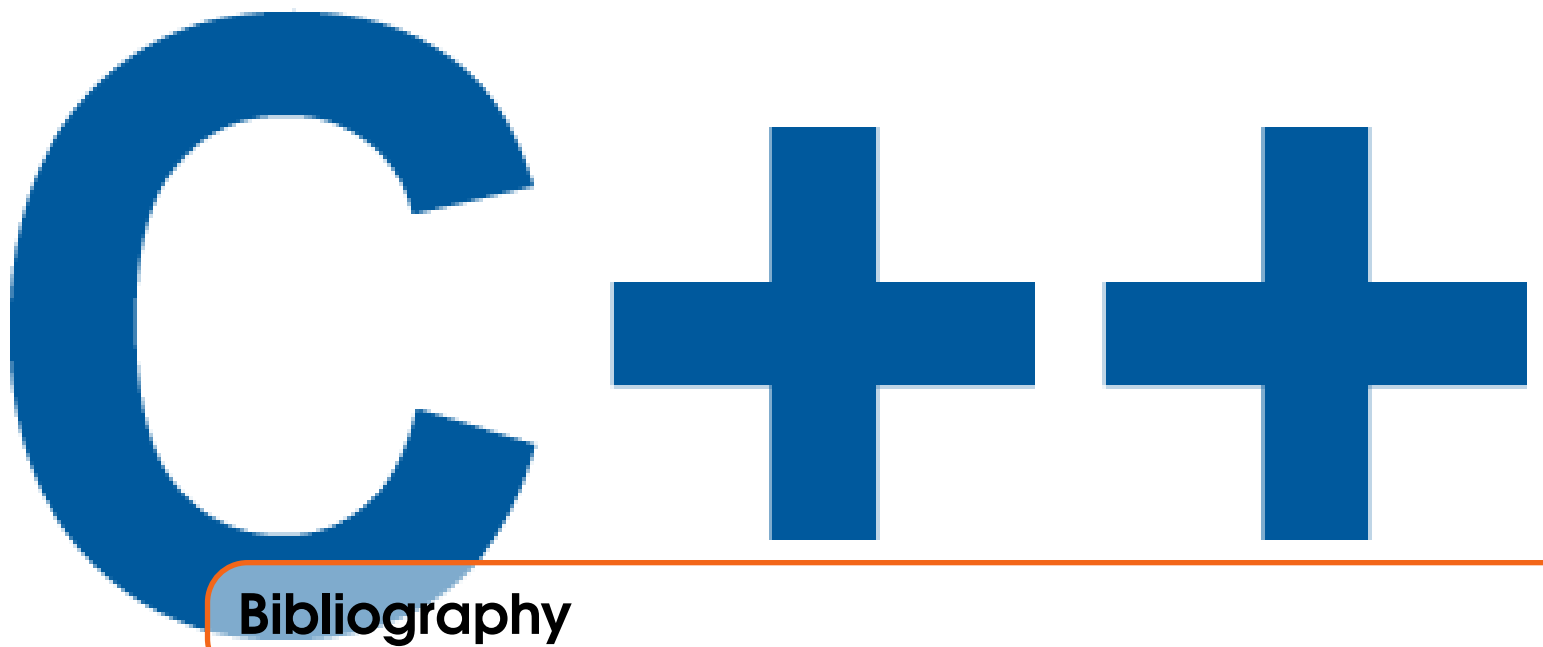
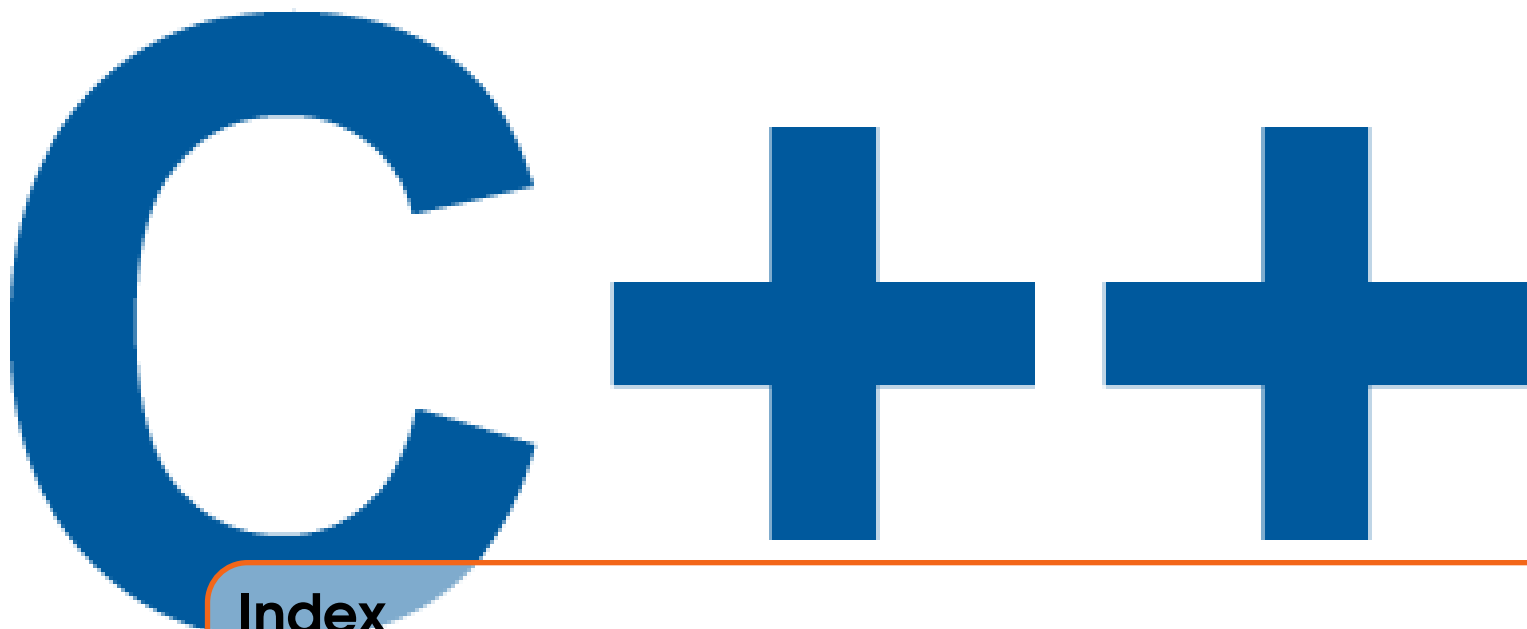


Figure 8.1: Figure caption



Books
Articles



Index

F

Figure.....31