Système d'Exploitation,

principes généraux et Shell de commandes

Licence d'Informatique

Gilles Dequen
gilles.dequen@u-picardie.fr

1

http://mis.u-picardie.fr/~dequen

2

Organisation du module

- Charte UPJV
  - Confidentialité des éléments de connexion
  - Usage réservé à la formation
  - Sauvegarde laissée à discrétion de l'utilisateur
- Conseils pour la réussite du module
  - Présence conseillée
  - Travail commun au plus en binôme
  - Travail personnel
    - Se convertir à l'usage des systèmes POSIX au quotidien
- Modalités de contrôle des connaissances
  - 3 examens (table et machines)

3

Plan

- Système d'exploitation
  - Terminologie
  - Principes généraux
- Système de Gestion de Fichiers
  - Organisation et commandes d'accès
- Le point de vue utilisateur
  - Maitrise du terminal
  - Droits d'accès, processus et jobs, …
- Ecriture de Shell-Script
- Maîtrise des expressions régulières
- Programmation système
  - Langage C

4

Principes généraux

- A propos de son identification sur le système (UPJV ou autre)
  - Chaque utilisateur est identifiable par un couple de données
    - son login (ou identifiant)
    - dans le système, un login est représenté par
      - UID : User IDentifier
      - GID : Group IDentifier
    - … et identifié par son password (ou mot de passe)
      - Règles de choix du password
        - Pas de mots courants (quelle que soit la langue)
        - Eviter date de naissance, numéros de téléphone, le « login », « azerty », « uiopqsd », …
        - Utilisation de minuscules, majuscules, chiffres et caractères de ponctuations (Attention au clavier !!)
          - Une méthode : prendre les premières lettres et/ou le codage SMS d'une phrase de référence.
  - "Crack", "Phishing", "Sniff", etc. strictement interdit

5

Principes généraux

- Procédure de connexion
- En mode texte ou graphique
  - Saisie du login
  - Authentification par le password
- Mode Texte
  - Accès à la console et démarrage d'un shell
  - Affichage de l'invite de commande prompt
  - Attente du shell pour la première commande
- Mode Graphique
  - Démarrage de l'interface graphique
  - Accès aux différents menu et applications grâce à la souris/clavier (shortcuts)

6

Principes généraux

- Lorsque la session est active
  - Accès à votre espace de stockage « privé » (OS Multi-Utilisateurs)
  - Cet espace est limité en taille
  - Un autre étudiant ne peut accéder à vos fichiers sans votre autorisation
    - Conséquence : vous ne pouvez pas accéder aux fichiers d'un autre étudiants
- Règle: vous ne devez pas rendre « publique » votre espace privé (cf Charte Informatique)
- Qui peut accéder aux espaces « privés » ?
  - le super-utilisateur (ou root)
  - Terme à l'origine du rootage des OS mobiles (Android, iOS/jailbreaké, . . .)
  - Il a le droit de tout faire sur le système (UID = 0)
  - Il a aussi la possibilité de faire n'importe quoi ! !
  - Privilège interdit en Licence, autorisé sous conditions à partir du Master. A tester chez vous le cas échéant !!

7

Principes généraux

- Procédure de déconnexion
- En mode texte ou graphique
  - Changer de session ou de terminal ne vaut pas déconnexion
  - La déconnexion est obligatoire à chaque fin de session (à l'UPJV)
- Mode Texte
  - Tapez la commande exit ou logout
  - shortcut Ctrl + d
  - retour à l'invite de login
- Mode Graphique
  - Dans le menu de session sélectionner « fermer la session »
  - Retour à la fenêtre de login

8

RAPPELS

9

Rappel: un « OS »

- O.S. = « operating system »

« An operating system (O.S.) is a set of softwares that manages computer hardware resources and provides common services for computer programs »

10

Rappel: un « OS »

- Terminologie
  - Processus
    - Instance de logiciel en cours d'exécution
    - Chaque processus possède son propre contexte d'exécution et donc son propre adressage mémoire
  - Thread
    - Processus dit « léger » ayant une portion de mémoire commune dite « partagée »
    - Rmq: un processus lourds peut se composer de plusieurs threads

11

Rappel: un « OS »

- O.S. = « operating system »
  - Caractéristiques d'un O.S.
    - Multi-processing
      - Gestion conjointe de plusieurs unités (physiques) de calcul
    - Multi-threading
      - Attention: différent de « multi-tâches »
    - Mono-utilisateur, Multi-utilisateurs
      - Notion de droits d'accès
    - Mono-tâche, Multi-tâches
      - Gestion conjointe de plusieurs processus lourds (ou « tâches »)

12

Rappel: un « OS »

- Questions ?

  - Peut-on être mono-tâche et multi-threadé ?
  - Peut-on être mono-utilisateur et multi-processing ?
  - Peut-on être mono-processeur et multi-threadé ?
  - Peut-on être mono-processeur et multi-tâches ?

13

Le Shell de Commandes

- Interface de commandes au clavier
  - Origine des interfaces de commandes
  - Permet le traitement par lots
    - Automatisation des tâches
    - Shell-Scripting
- Déclinaison des Shells de Commandes UNIX
  - Bourne-Shell: bash (ou sh)
  - C-Shell, csh ou tcsh
  - Perl, Python

14

Le Shell de Commandes

- Contrôle des Processus
- Déplacement et manipulation du Système de Gestion de Fichiers
- Langage de Contrôle
  - Manipulation de Variables
    - Parfois sommaire
  - Contrôle de Flux et des Services
  - Expressions régulières

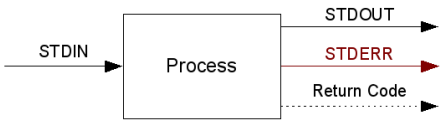
15

Le Shell de Commandes: E/S

- Sous UNIX
  - Tout est considéré comme « fichier »
  - Concept des Entrées/Sorties Standard
    - Un programme consomme un flux d'entrée
      - Par défaut le Clavier (nommé stdin)
    - Produit un résultat sur un flux de sortie
      - Par défaut le terminal (nommé stdout)
    - Transmet ses messages d'erreur sur un flux d'erreurs
      - Par défaut le terminal (nommé stderr)

16

Le Shell de Commandes: E/S



17

Le Shell de Commandes

- Syntaxe Générale d'une Commande
  - Convention: % = Invite de commande
  - % cmd [options] … [arguments]
  - cmd : le nom de la commande
  - [options]
    - Chaines de caractères optionnelles permettant de modifier le comportement de cmd
    - Autre nom: les Paramètres
  - [arguments]
    - Spécifie les données sur lesquelles s'applique cmd
- Remarque: valable en mode interactif ou en traitement par lots

18

Le Shell de Commandes

- Syntaxe Générale d'une Commande
  - Convention: % = Invite de commande
  - % cmd [options] … [arguments]
  - cmd : le nom de la commande

```
% cmd1 ↵          ≅   % cmd1 ; cmd2; cmd3; cmd4 ↵
% cmd2 ↵
% cmd3 ↵
% cmd4 ↵
```

19

Le Shell de Commandes

- Première commande utilisée
  - % passwd
  - Pas d'options
  - Pas d'arguments
    - Fonction déclenchant la procédure de changement du mot de passe de l'utilisateur courant

20

sh: La documentation en Ligne

% man [section] cmd

```
Section :
   1) Users commands     (ex: man 1 ls)
   2) System calls             (ex: man 2 chmod)
   3) Subroutines        (ex: man 3 printf)
   4) Devices                  (ex: man 4 NFS)
   5) File formats             (ex: man 5 passwd)
   6) Games                    (ex: man 6 chess)
   7) Miscellaneous            (ex: man 7 man)
   8) AdminSys           (ex: man 8 mkfs)
   9) Kernel                   (ex: man 9 ext2)
```

21

sh: La documentation en Ligne

- NAME
- SYNOPSIS
- AVAILIBILITY
- DESCRIPTION
- OPTIONS
- DEFAULTS
- EXAMPLES
- OVERVIEW
- ENVIRONMENT
- EXIT STATUS
- WARNINGS
- FILES
- SEE ALSO
- DIAGNOSTICS
- BUGS
- HISTORY

22

sh: La documentation en Ligne

```
PASSWD(1)                 BSD General Commands Manual                 PASSWD(1)
NAME
     passwd -- modify a user's password
SYNOPSIS
     passwd [-i infosystem [-l location]] [-u authname] [user]
DESCRIPTION
     The passwd utility changes the user's password.  If the user is not
the super-user, passwd first prompts for the current password and will not
continue unless the correct password is entered.  …
FILES
     /etc/master.passwd  The user database
     /etc/passwd         A Version 7 format password file
     /etc/passwd.XXXXXX  Temporary copy of the password file
SEE ALSO
     chpass(1), login(1), dscl(1), passwd(5), pwd_mkdb(8), vipw(8)
     Robert Morris and Ken Thompson, UNIX password security
HISTORY
     A passwd command appeared in Version 6 AT&T UNIX.
```

23

LE SYSTEME DE GESTION DE FICHIERS (SGF)

24

sh: Se déplacer dans le SGF

- Rappel sur l'arborescence du SGF
  - / : La racine du système (root)
  - . : Le dossier courant (signifiant « ici »)
  - .. : Le dossier parent (signifiant « au dessus »)
    - Tous les dossiers contiennent au moins « . » et « .. »
      - « . » et « .. » sont identiques à la racine
  - ~ : Signifie « chez moi »
  - ~user : Signifie « chez user »

25

sh: Se déplacer dans le SGF

- Rappel sur l'arborescence du SGF
  - Adressage
    - Tout adressage dans l'arborescence débutant par le caractère « / » est **absolu**
    - Il est **relatif** sinon

26

sh: le SGF

- Rappel sur l'arborescence « **UNIX** »
  - /home – dossier des utilisateurs
  - /bin, /usr/bin – commandes système
  - /sbin, /usr/sbin – commandes système « admin »
  - /etc – fichiers de configuration
  - /var – dossiers de logs, spooler, etc …
  - /dev – points d'entrée des périphériques
  - /proc – fichiers systèmes spéciaux

27

sh: Se déplacer dans le SGF

- Se repérer dans l'arborescence du SGF
  - `% pwd`
    - Envoie sur le flux de sortie standard l'adresse absolue du dossier courant
- Lister les entrées d'un dossier
  - `% ls [-aliARF …] [list of names]`
    - `-a` : « all », fichiers cachés compris
    - `-l` : « long », visualisation au format long
    - `-R` : « recursive mode »
  - `% man ls`
    - Pour le reste de la commande

28

sh: Se déplacer dans le SGF

- Se déplacer dans le SGF

`% cd [dir]`

- Signifiant « change directory »
- « Se place » dans le dossier passé en paramètre
  - Modification du « dossier courant »
  - Condition: l'adressage doit être valide. Retour erreur sinon.
- Si absence de paramètre
  - Se place dans le dossier « ~ »

29

sh: Se déplacer dans le SGF

- Création de branches dans l'arborescence
  - `% mkdir [-p] dirname`
    - Signifie « make directory »
    - Création de dossier
    - Option « -p »
      - Création de branches intermédiaires si nécessaire
- Suppression de dossier
  - `% rmdir dirname`
    - `dirname` doit être vide
      - Voir l'option « recursive » de la commande `rm`

30

sh: Se déplacer dans le SGF

- Création de fichiers (d'entrées)

  % touch [-A date] filename

  - Si filename n'existe pas
    - Crée le fichier filename
      - « droits par défaut » positionnés
  - Si filename existe
    - Altération des dates d'accès

31

sh: Les fichiers et l'arborescence

% ls -al

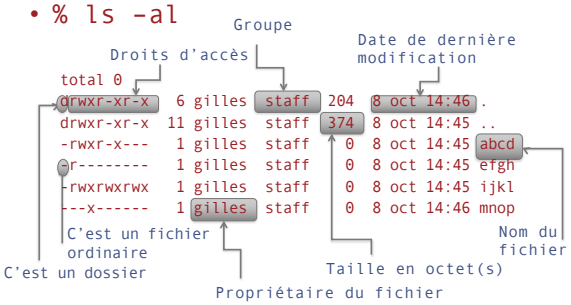
- -l : Affichage au format « long »
- -a : Affichage des fichiers cachés
  - Un fichier est caché si son nom débute par un « . »

```
total 0
drwxr-xr-x   6 gilles  staff  204  8 oct 14:46 .
drwxr-xr-x  11 gilles  staff  374  8 oct 14:45 ..
-rwxr-x---   1 gilles  staff    0  8 oct 14:45 aaaa
-r--------   1 gilles  staff    0  8 oct 14:45 bbbb
-rwxrwxrwx   1 gilles  staff    0  8 oct 14:45 cccc
---x------   1 gilles  staff    0  8 oct 14:46 dddd
```

32

sh: Les fichiers et l'arborescence

- % ls -al



33

sh: Les fichiers et l'arborescence

- Un fichier du Système de Gestion de Fichiers
  - Associé à exactement un i-noeud
    - Et un i-noeud est associé à excatement un fichier
  - Et un i-nombre
    - Pas de nom dans le i-noeud
- i-nœud
  - Propriétaire : le UID
  - Groupe: le GID
  - Dates, droits d'accès, nombre de liens, taille, type du fichier

34

sh: Les fichiers et l'arborescence

- Le lien
  - Système de Gestion des Fichiers
    - Structuration arborescente des fichiers
    - Conséquence: pas de cycle
      - Propriété intrinsèque de la structure d'arbre (Cf. L2 Info)
  - Lien
    - Rupture de l'arborescence
    - Lien Symbolique
      - un fichier qui « pointe » un autre fichier de l'arborescence
      - Ex: un raccourci Windows
    - Lien Physique
      - une expression multiple d'un même i-noeud

35

sh: Les fichiers et l'arborescence

- Le lien

  % ln [-s] {source} {destination}
  - Modification du SGF en vue d'avoir plusieurs références aux même données
  - Lien par le nom
    - Un lien « symbolique »
      - option « -s »
  - Lien par le i-nœud
    - Un lien « physique »

36

sh: Les fichiers et l'arborescence

- Le lien « symbolique »
  % ln -s {source} {destination}
  - C'est un alias
    - Un lien par le nom
    - Fichier « pointant » un autre fichier
      - Conséquence: une nouvelle entrée dans la table des i-noeuds
    - Cas des « raccourcis windows »
  - Usage courant
    - Passage au travers des frontières physiques
  - Pas de droits sur le lien
    - Les droits sont ceux du fichier/dossier lié

37

sh: Les fichiers et l'arborescence

- Le lien « physique »
  % ln {source} {destination}
  - Le même i-nœud est adressable à différents endroits de l' arborescence
    - Conséquences:
      - Pas de nouvelle entrée dans la table des i-noeuds
      - Pas de passage au travers des frontières physiques
  - La « source » et la « destination » sont donc un seul et même fichier

38

sh: Les fichiers et l'arborescence

- Fichier (informatique)
  - Ensemble d'informations binaires décrivant un objet numérique
    - Musique, Image, Vidéo, Texte, …
  - Il appartient nécessairement à un dossier
  - Il a un nom
    - Le nombre de caractères maximal nommant un fichier est fixé par le SGF
  - Il appartient à quelqu'un
    - Droits d'accès

39

sh: Les noms de fichiers

- Nom de fichier
  - Chaine de caractères finie
    - Caractères interdits
      / * ? [ ] < > $ ’ ” & !
  - Aucune propriété héritée par le nom de fichier
  - Extension de fichiers
    - Une convention (rien de plus)
    - Une chaine de caractère « concaténée » au nom et séparée par un « . »

40

sh: Les noms de fichiers

- Nom de fichier
- Extensions de fichiers

.doc	.pdf	.exe
.xls	.tex	.sh
.java	.zip	.7z
.c	.svg	.bak
.class	.gz	.tar
.jpg	.rar	.wav
.html	.bz2	.mp3

41

sh: Les noms de fichiers

- Metacaractères et Noms de Fichiers
  - Caractères spéciaux interprétés par le shell
    - La substitution des metacaractères se fait avant l’exécution de la commande
  - But
    - Spécifier des ensembles de fichiers

? : Un caractère quelconque

[ ] : Un caractère appartenant à l’ensemble entre []

* : N’importe quelle suite de caractères (y compris vide)

\ : Masque. Le caractère qui le suit ne sera pas substitué le cas échéant

42

sh: Les noms de fichiers

- Metacaractères
  - « ? »
    - Un caractère quelconque
  - Exemple

```
% ls
abcd  bacd  cbcd  bbcd  azbc
% ls ?bcd
abcd  cbcd  bbcd
% ls b?cd
bacd  bbcd
```

43

sh: Les noms de fichiers

- Metacaractères
  - « [ ] »
    - Un caractère appartenant à l'ensemble entre []
  - Exemple

```
% ls
abcd  bacd  cbcd  bbcd  azcd
% ls [ab]bcd
abcd  bbcd
% ls [a-c]bcd     // [a-c] signifie « de 'a' à 'c' »
abcd  cbcd  bbcd
% ls [ab][az]cd
bacd azcd
```

44

sh: Les noms de fichiers

- Metacaractères
  - « * »
    - N'importe quelle suite de caractères (y compris vide)
  - Exemple

```
% ls
abcd  bacd  cbcd  bbcd  azbc
% ls *cd
abcd  bacd  cbcd bbcd
% ls a*c
azbc
% ls *
abcd  bacd  cbcd  bbcd  azbc
```

45

sh: Les noms de fichiers

- Identifier le « type » d'un fichier
  - Par l'extension ?
    - Rappel: elle ne conditionne rien. C'est une convention.
  - Stéganographie
    - Dissimulation d'informations

```
% file file_1 file_2 … file_n
```

  - Identification par le « nombre magique »
  - Exemple

```
% file vallst_0.9.258.tar
vallst_0.9.258.tar: POSIX tar archive (GNU)
```

46

Les droits d'accès

47

sh: Les droits d'accès

```
% ls -al
total 0
drwxr-xr-x  11 gilles  staff  374  8 oct 14:45 ..
-rwxr-x---   1 gilles  staff    0  8 oct 14:45 aaaa
lrwxrwxrwx   1 gilles  staff    0  8 oct 14:45 cccc
```

- Le premier caractère: Le type du fichier
  - « - » : fichier ordinaire
  - « d » : dossier
  - « l » : lien symbolique
  - D'autres sont possibles …

48

sh: Les droits d'accès

```
% ls -al
total 0
drwxr-xr-x  11 gilles  staff  374  8 oct 14:45 ..
-rwxr-x---   1 gilles  staff    0  8 oct 14:45 aaaa
lrwxrwxrwx   1 gilles  staff    0  8 oct 14:45 cccc
```

- Caractères 2 à 4 : Droits d'accès du propriétaire
  - « r » ou « - » : «**R**ead Access» ou non
  - « w » ou « - » : «**W**rite Access» ou non
  - « x » ou « - » : «e**X**ecute ability» ou non

49

sh: Les droits d'accès

```
% ls -al
total 0
drwxr-xr-x  11 gilles  staff  374  8 oct 14:45 ..
-rwxr-x---   1 gilles  staff    0  8 oct 14:45 aaaa
lrwxrwxrwx   1 gilles  staff    0  8 oct 14:45 cccc
```

- Caractères 5 à 7 : Droits d'accès du groupe
- Caractères 8 à 10 : Droits d'accès des autres
- En conclusion, 3 triplets de lettres définissent les droits d'accès

50

sh: Les droits d'accès

- Droits sur les Fichiers
  - **R**ead
    - **Droit de lire.** « Je peux éditer/visualiser mon fichier »
  - **W**rite
    - **Droit de modifier.** « Je peux écrire dans le fichier et l'effacer »
  - e**X**ecute
    - **Droit d'exécuter.** « S'il contient des instructions, je peux les exécuter »
    - Rmq: Pour exécuter un ordre, il faut pouvoir prendre connaissance de l'ordre. Pas de « **x** » si pas de « **r** »

51

sh: Les droits d'accès

- Droits sur les Dossiers
  - **R**ead
    - **Droit de lire.** « Je peux lister/visualiser les fichiers qu'il contient »
  - **W**rite
    - **Droit de modifier.** « Je peux créer et supprimer des fichiers du dossier »
  - e**X**ecute
    - **Droit d'exécuter.** « Je peux me positionner dans le dossier ou le traverser. C'est un droit de passage »
    - Rmq: Pour pouvoir disposer de « **r** » et/ou « **w** », il faut « **x** »

52

sh: Les droits d'accès

- Le changement des droits
  - Mode « **Symbolique** »

```
% chmod [-R] [ugoa][+-=][rwxstugo] file …
```

  - -R : récursivité de la commande
  - ugoa : **u**ser, **g**roup, **o**thers or **a**ll
  - +-= : ajout, retrait ou positionnement
  - rwx : **r**ead, **w**rite ou e**x**ecute
  - stT : set-uid, set-gid ou sticky
  - ugo : recopie des droits user, group ou others

53

sh: Les droits d'accès

- Le changement des droits
  - Mode « **Symbolique** »
  - Exemple

```
% ls -l file2
-rw-r--r--   1 gilles  staff  38237 file2
% chmod ug+x file2
% ls -l file2
-rwxr-xr--   1 gilles  staff  38237 file2
```

54

sh: Les droits d'accès

- Le changement des droits
  - Mode « **octal** »

```
% chmod [-R] <octal value> file …
```

  - 3 (ou 4) chiffres octaux à sommer
  - 4000 set-IUD bit
  - 2000 set-GID bit
  - 1000 sticky-bit

  } droits « **spéciaux** »

- 400 : droit « u+r » 40 :droit « g+r » 4 : droit « o+r »
- 200 : droit « u+w » 20 :droit « g+w » 2 : droit « o+w »
- 100 : droit « u+x » 10 :droit « g+x » 1 : droit « o+x »

55

sh: Les droits d'accès

- Le changement des droits
  - Mode « **octal** » - **En Pratique**
  - Chaque triplet est représenté par sa traduction en octal
  - Modification « **absolue** » du droit

rwx r-x r--

111 101 $100_{(2)}$

7 5 $4_{(8)}$

56

sh: Les droits d'accès

- Le changement des droits
  - Mode « **octal** »
  - Exemple

```
% ls -l file2
-rw-r--r--   1 gilles  staff  38237 file2
% chmod 750 file2
% ls -l file2
-rwxr-x---   1 gilles  staff  38237 file2
```

57

sh: Les droits d'accès

- Les droits par défaut dit « maximaux »
  - Création d'un fichier « myfile »

    `% touch myfile`
  - Quels sont les droits maximaux à la création d'un fichier ?

    `-rw- rw- rw-`

    $000\ 110\ 110\ 110_{(2)}$

    $0\quad 6\quad 6\quad 6_{(8)}$
    - **Remarque:** Pas de droit d'exécution positionné **par défaut** lors de la création d'un fichier
      - Exception pour l'**édition de liens**

58

sh: Les droits d'accès

- Les droits par défaut dit « maximaux »
  - Création d'un dossier « mydir »

    `% mkdir mydir`
  - Quels sont les droits à la création d'un dossier ?

    `-rwx rwx rwx`

    $000\ 111\ 111\ 111_{(2)}$

    $0\quad 7\quad 7\quad 7_{(8)}$

59

sh: Les droits d'accès

- Détermination du droit par défaut
  - Opération de masquage
  - ET binaire
    - Noté « & »

&	0	1
0	0	0
1	0	1

  - NOT unaire (complément)
    - Noté « ~ »

	~
0	1
1	0

60

sh: Les droits d'accès

- Détermination du droit par défaut
  - Règle de calcul

  « Droits Maximaux » & ~MASK
  - Exemple

    MASK = $022_{(8)}$ = $000\ 010\ 010_{(2)}$

    % mkdir mydir ## création d'un dossier

```
        111 111 111 & ~(000 010 010)
Et donc 111 111 111 &   111 101 101
Soit    111 101 101 par défaut
```

61

sh: Les droits d'accès

- Les droits par défaut
  - Changement des droits par défaut POSIX

  % umask [[x]xxx]
  - Défini le masque du droits par défaut: **MASK**
    - Affichage du droit par défaut si l'argument n'est pas précisé
  - Exemple

    % umask 022

62

sh: Les droits d'accès

- Les droits par défaut
  - Exercice: Quels seront les droits des entrées suivantes

```
%umask 000 ; touch file1 ⇒ rw- rw- rw-
%umask 033 ; mkdir dir1  ⇒ rwx r-- r--
%umask 077 ; touch file2 ⇒ rw- --- ---
%umask 027 ; mkdir dir2  ⇒ rwx r-x ---
%umask 447 ; touch file3 ⇒ -w- -w- ---
%umask 277 ; mkdir dir3  ⇒ r-x --- ---
%umask 111 ; mkdir dir4  ⇒ rw- rw- rw-
```

63