Zapping informatique : présentation rapide de concepts fondamentaux

Antoine AMARILLI

1 Système de contrôle de versions : travail collaboratif et documents versionnés

Problème : suivi des modifications et travail collaboratif.
- Architecture : dépôt (à voir comme une abstraction) et copies de travail, commits versionnés.
- Accès au dépôt : sur la même machine (décentralisés), par Internet, voire par le WWW (lien avec Wikipédia).
- Conflits et verrous.
- Usage disque (cheap copies, diffs).

2 Différences entre deux fichiers et sommes de contrôle

Problème : vérifier si deux fichiers sont différents, et trouver les différences. (Pas la taille.)
- Sommes de contrôle : exemple du bit de parité, lien avec la transmission Internet et limites. Permet une comparaison rapide.
- Hachage cryptographique MD5 : somme non réversible. Utilisation pour signer (pantoufle de Cendrillon).
- Diff.
- Patch. (F.C.)
- Lien avec les SCV.

3 Cryptographie

Problème : Internet, les e-mails, les données sur les disques, sont en clair.
- Crypto symétrique : même clé pour coder et décoder.
- Principe de la crypto asymétrique (et aspect paradoxal). Lien avec les nombres premiers.
- Signature sécurisée.
- Signature de clés et réseaux de confiance.

4 Texte marqué et séparation forme/contenu

Problème : Difficile de mettre en page de gros documents de manière cohérente et propre (support des équations, par exemple...).
- Principe du WYSIWYM et avantages (meilleur contrôle sur la mise en page (tout sous les yeux), ne pas interrompre la rédaction, pas de souris, pas besoin d'un logiciel spécifique (juste un bon éditeur de texte) ou d'un ordinateur puissant).
- Divers langages : HTML, LaTeX, etc.
- Séparation forme-contenu (pas vraiment liée à l'aspect précédent mais...) : marquage sémantique et feuille de style. Meilleure homogénéité et permet d'avoir plusieurs feuilles de style (différents médias, différents styles).
- Meilleure accessibilité.
- LaTeX : commandes (héritage, structure (plusieurs paramètres)), HTML : sélecteurs (détection de l'imbrication, "ou" logique, before et after (sémantique perdue, mais DTD ?)).

5 XML, DTD et XSLT

Problème : formats de fichier incompatibles. (Exemple : HTML et LaTeX)
- XML : métaformat (définit le format des formats).
- Différents dialectes : SVG, XHTML, ODF, DocBook, le vôtre... Exemple : pour un cours de mathématiques.
- Langage de description de format de document : DTD (pas XML) et XML Schema. Vérifie que le document est valide (suit les règles de formation), à différencier de "bien formé". Exemple : identifier que les éléments "démonstration", "corollaire" sont dans le désordre.
- Langage de transformation : XSLT (lien avec les feuilles de style). Permet de transformer le document en autre chose. Exemple : page web, LaTeX... (Lui-même du XML, comme Schema !)

6 Scripts make et compilation

Problème : Comment gérer plusieurs formats d'un même document de manière intelligente ?
- Gérer les sources en un seul exemplaire et décrire comment les transformer en différents formats compilés. (Pas de copier-coller !)
- Gestion de cibles.
- Dépendances.
- Règles d'inférence.
- Lien avec bitmap vs. vectoriel.

7 Système de gestion de paquets : dépendances et automatisation

Problème : Il est long d'installer beaucoup de logiciels.
- Dépôts de paquets centralisés (rien à voir avec les SCV). Téléchargement et installation automatisés.
- Permet la mise à jour automatique.
- Gestion des dépendances (pour les bibliothèques notamment).
- Permet une sauvegarde facile de la liste des applications installées. Réplication sur un réseau.

8 Répertoire utilisateur et droits d'accès

Problème : Les données des utilisateurs ne sont pas regroupées et certaines sont partagées (exemple pour Windows, backup).
- Il suffit de les centraliser dans un répertoire pour chacun. Interdiction aux applications d'écrire ailleurs : chacun a ses paramètres. (Schéma.)
- Comment installer de nouvelles applications ? Droits d'administration. Permet une meilleure sécurité (un utilisateur ne peut pas mettre en danger les autres).
- Gestion plus fine des droits de chacun : système de permissions.

9 Bases de sécurité informatique

Problème : Les ordinateurs ne sont pas sécurisés.
- Aspect déterministe de la sécurité (par opposition aux idées préconçues) : un programme obéit à son code (sur un matériel parfait, pas de sécurité matérielle).
- Description haut niveau du programme. Conflit avec le comportement du programme (avec le code) si auteur malicieux, bug ou faille de sécurité.
- Sécurité par l'obscurité ? Cas limite : pas besoin de cacher le code s'il n'y a pas de bogues.
- Attaques sur un système : confidentialité (keyloggers, spyware), intégrité (destruction ou falsification : bulletins), ou disponibilité (stockage, vers qui se répliquent, botnets et backdoors). Attaques dues à un programme, éventuellement avec intervention humaine synchronisée (cracking).
- Compartimentation : l'OS a tous les droits, les programmes-fils ont des droits plus petits ou égaux, gestion des accès à différents types de ressource (bande passante, mémoire disque, puissance de calcul). Pour accéder à des données : élévation des privilèges. Nécessité de cloisonner par rapport à la description haut niveau. Ce qui se passe quand on exécute un virus.

- Authentification (mots de passe et autres), social engineering, shoulder surfing (exemple).
- Antivirus : base de signatures de programmes malveillants (mais définition ? [RJL software], collisions ? pas exhaustif), analyse heuristique (en fait gestion des droits palliant un manque de l'OS ou de l'utilisateur [mais bonne idée de surveiller les couches-filles), et corrige l'OS (pas son boulot : et s'il se fait compromettre).
- Pare-feu : gestion des accès réseaux (entrants et sortants).
- Conclusion : antivirus et pare-feu sont en fait une gestion des droits des programmes (et utilisateurs).

10 SSH

Problème : On veut pouvoir accéder à son ordinateur à distance.
- La console est une interface moins gourmande en ressources.
- SSH permet d'ouvrir une session à distance. (Pas un outil de cracker en soi !)
- Cryptage.
- Administration de serveurs à distance.
- Idée de tunnels, sur connexion non sécurisée (exemple).

Conclusion

- Espère que ce fut enrichissant.
- Questions (s'il y a le temps) ou renvoi au repas de classe.

Ordre de passage

Groupe 1

1. — 4 [10 min]
2. — 1 [10 min]
3. — 2 [10 min]
4. — 3 [5 min]
5. — 9 [5 min]
6. — 10 [5 min]

Groupe 2

1. — 7 [5 min]
2. — 8 [5 min]
3. — 9 [10 min]
4. — 4 [10 min]
5. — 5 [7 min]
6. — 6 [7 min]