

UNIVERSITE DE KINHASA



FACULTE DES SCIENCES ET TECHNOLOGIE

Mention Statistiques et Informatique

B.P. 190 KINHASA XI



SYSTEME D'EXPLOITATION

Projet : Expressions Régulières et Utilitaires

GRP- 06

Par

DINGENO	MUNUMA	OLIVIER
BINAKI	KUETUENDA	ANDY
BABUTEMBE	ABOVA	PATRICIA
MAYEYE	SUNDANI	GEMIMA
IZUNGU	BAKUMBONGO	
MVI	TOMY	FLORENT
BANANI	NGINDU	DANY
BANGONGO	LUKOMBO	MERDI
MWELA	KIMBINZI	DON
GITADI	GIBANZALA	PETER

Tous en L2 LMD

Titulaire du cours : Prof. KASENGEDIA MOTUMBE Pierre

Collaborateurs :

Junior KANINGINI & Fipa BUKUSU & Jacques MAYOMBO

Présentation

**Notre travail consiste à parler profondément sur les
Expressions régulière et utilitaires dans notre cours
intitulé SYSTÈME D'EXPLOITATION.**

RAPPORT

PROJET SYSTEME D'EXPLOITATION

1. Expressions régulières :

Elles sont le moyen simple d'écrire des séquences des caractères. Depuis continuellement en informatique, la quête de caractères donnés ou de séquences de caractères concrètes dans des documents fait partie des tâches standards dont il faut régulièrement s'acquitter. L'objectif dans ce cas est souvent de modifier ou de remplacer des portions de texte ou des lignes de code.

Cette tâche est d'autant plus complexe que la séquence de caractères quêtée apparaît plusieurs fois dans le document. Dans les années 1950 déjà, les experts avaient trouvé dans les langages de l'information théorique une solution qui marque dorénavant encore l'épanouissement de logiciels et simplifie largement ce type de tâches répétitives sous la forme d'expressions régulières.

Les Expressions régulières sont les unités de description de langages rationnels faisant partie des langages formels. Elles sont un instrument central de l'informatique théorique, qui forme notamment le soubassement du progrès et de l'exécution des programmes informatiques ainsi que de la construction des compilateurs requis à cet effet. C'est la raison pour laquelle les expressions régulières (souvent également qualifiées de REGEX et reposant sur des règles syntaxiques clairement définies) sont notamment utilisées dans le développement de logiciels.

Il existe pour chaque expression régulière un automate fini (également qualifié d'automate avec un nombre fini d'étapes) qui accepte un langage spécifié par l'expression et qui est développé à l'aide de la construction de Thompson à partir d'une expression régulière. Il existe qui plus est pour chaque automate fini

une expression régulière qui décrit le langage accepté par l'automate. Celle-ci peut être créée soit par l'algorithme de Kleene soit par élimination d'états.

NB : un automate est un modèle comportemental se composant d'états, de transitions d'états et d'actions. Il est qualifié de fini lorsque la quantité d'états qu'il peut accepter est finie (autrement dit limitée).

Un exemple connu de l'utilisation des expressions régulières est la fonction rechercher/remplacer des éditeurs de texte que le pionnier en informatique Ken Thompson l'un des développeurs du système d'exploitation UNIX a mis en place pour la première fois dans les années 1960 dans l'éditeur orienté ligne QED, puis dans son successeur ED.

Cette fonction permet de rechercher des séquences de caractères données dans des textes et de les remplacer éventuellement par une autre séquence de caractères quelconque.

Par définition Les Expressions Régulières sont des chaînes de caractères sur la base de règles syntaxiques permettant de décrire des séquences de caractères. Elles font de fait partie des langages rationnels.

Fonctionnement d'une expression régulière :

Une expression régulière peut-être composée soit exclusivement de caractères normaux (par ex : abc) soit d'une combinaison de caractères normaux et de Meta caractères (par ex : ab*c). Le but des Meta caractères ici est de décrire des constructions ou agencements de caractères donnés, par exemple de décrire si un caractère doit se trouver au début de la ligne, s'il doit ou peut n'apparaître qu'une fois ou plus ou moins souvent. Les deux exemples d'expressions régulières cités fonctionnent ainsi de la manière suivante :

abc : le modèle simple de REGEX abc requiert une correspondance exacte. La recherche s'attachera donc à détecter des chaînes de caractères contenant non seulement tous les caractères «abc», mais faisant également apparaître dans cet ordre précis. La correspondance demandée par l'expression se trouve donc dans

la question « connais-tu l'abc ? » comme dans la phrase « je dois encore soigner mon abcès ».

ab*c: Des expressions régulières avec des caractères spéciaux fonctionnent en revanche un peu différemment, car la recherche se fait non seulement au niveau de correspondances exactes, mais aussi de scénarios spéciaux. L'étoile (également appelée « astérisque ») dans ce cas veille à rechercher de chaînes de caractères commençant par la lettre « a » et finissant par la lettre « c », mais pouvant avoir entre les deux un nombre quelconque de lettres « b ». Il y aura donc une correspondance non seulement dans « abc », mais aussi dans les chaînes de caractères « abbbbc » et « cbbabbcba ».

Les enjeux liés à l'emploi d'expression régulière :

Toute personne travaillant avec des instructions de REGEX à une grande liberté, car il y a toujours plusieurs options de solution pour chaque mission devant être résolue avec une expression régulière. Il n'est néanmoins pas toujours avantageux qu'un résultat souhaité puisse être obtenu de différentes manières :

Vous pouvez par exemple garder les instructions à un niveau très général pour atteindre l'objectif visé quoi qu'il arrive. Si vous voulez en revanche le résultat le plus précis possible, vous devez impérativement formuler un modèle de REGEX spécifique. Il est également conseillé de faire attention à la longueur : plus une expression régulière est concise, plus sa durée de traitement est courte. Elle doit néanmoins toujours être lisible. Car si vous souhaitez modifier ultérieurement les expressions régulières utilisées, vous devrez faire face à un obstacle de taille si les instructions initiales sont trop compliquées, et qui plus est sans commentaire.

Lors de la création d'expressions régulières, il s'agit donc de trouver un rapport optimal entre la concision et la spécificité.

Les règles syntaxiques qui s'appliquent aux expressions régulières :

Comme indiqué précédemment, les expressions régulières peuvent être employées dans divers langages (par exemple **Perl**, **python**, **Ruby**, **JavaScript**, **XML** ou **HTML**).

Une expression régulière peut ainsi se diviser en trois parties au maximum, indépendamment du langage dans lequel elle est utilisée : **Motif, Délimiteur, Modificateur.**

Sortes d'expressions régulières :

- **Expressions régulières à un élément :**

La forme la plus simple d'expression régulière est un modèle de recherche à un seul élément. Puisque vous ne recherchez pas un élément spécifique, vous pouvez facilement définir une expression régulière pour un élément en utilisant **une classe de caractères**, par exemple. Dans l'expression suivante, les résultats possibles sont des chiffres

1 2 3 4 5 6 7 [1 2 3 4 5 6 7]

Dans ce cas, les nombres se suivent directement, donc la forme simplifiée suivante est également possible :

[1-7]

Une version simplifiée avec un signe moins est également disponible si l'expression régulière est modifiée pour exclure le chiffre "4" de la recherche, vous pouvez également utiliser la variante simplifiée avec le signe moins :

[1-35-7]

- **Expressions régulières à plusieurs éléments**

Même pour les expressions régulières avec plusieurs éléments, vous pouvez manipuler des classes de caractères pour obtenir des résultats différents. Par exemple, si une expression contient deux éléments avec des résultats possibles différents, vous pouvez simplement concaténer les deux classes de caractères correspondantes **l'une à la suite de l'autre** :

[1-7] [a-c]

Donc soit la lettre 'a', 'b' ou 'c' suit le premier élément. C'est un nombre de '1' à '7'. Comme mentionné ci-dessus, les minuscules sont obligatoires. Avant d'entrer dans les modificateurs ici, nous pouvons déjà aborder le petit changement suivant dans l'expression "contenir des majuscules".

[1-7] [a-cA-C]

- **Expressions régulières avec des éléments en option**

Que vous recherchiez plusieurs éléments dans une seule expression régulière ou que vous utilisiez plusieurs ensembles de caractères, il est possible que les entrées données puissent ou ne doivent être contenues que sous certaines conditions. Par exemple, cela peut se produire dans le cas d'une expression régulière qui doit filtrer tous les numéros de maison. Les numéros de maison se composent parfois d'un seul chiffre, mais les résultats incluent également des nombres composés de deux ou même trois chiffres. Il existe également des adresses où une lettre est ajoutée au numéro de maison. Cet ensemble de combinaisons possibles peut idéalement être couvert par les instructions d'expression régulière suivantes :

[1-9] [0-9] ? [0-9] ? [a-z ?]

Le seul élément obligatoire de ce formulaire de recherche est un chiffre entre « 1 » et « 9 ». Deux chiffres entre "0" et "9" suivis d'une lettre quelconque peuvent être sélectionnés comme options, respectivement caractérisés par un point d'interrogation suivi.

Bien qu'il soit encore assez évident de construire des nombres à trois chiffres suivis d'une lettre supplémentaire, ce n'est pas vrai pour les nombres à dix chiffres. Dans ce cas, des accolades doivent être utilisées, comme dans l'exemple d'expression régulière suivant :

[1-9] [0-9] {0,9}

Comme dans l'exemple précédent, la recherche se concentre d'abord sur un seul chiffre entre « 1 » et « 9 », puis ne recherche pas ou jusqu'à neuf chiffres de « 0 » à « 9 ». Par conséquent, le résultat peut comporter jusqu'à dix chiffres.

- **Expressions régulières avec n'importe quelle fréquence des répétitions.**

Dans les exemples précédents d'expressions à un seul élément et à plusieurs éléments, le nombre minimum et maximum de caractères était connu.

Cependant, bien sûr, il existe des cas où le nombre de caractères dans l'expression régulière n'a pas besoin d'être déterminé exactement à l'avance. Ensuite, les paramètres requis sont l'astérisque et le signe plus, permettant n'importe quel nombre de répétitions d'un caractère, d'une classe ou d'un groupe de caractères. Ainsi, tous les littéraux de chaîne peuvent être entrés avec n'importe quel nombre de chiffres (même "zéro"), par exemple avec l'expression régulière suivante :

[0-9]*

2. Utilitaires :

Les logiciels utilitaires sont de logiciels applicatifs, qui permettent à l'utilisateur d'effectuer des manipulations basiques telles que démarrer.

Un logiciel utilitaire (aussi appelé programme utilitaire ou simplement **utilitaire**) est un logiciel conçu pour analyser, configurer, optimiser ou entretenir une pièce d'équipement informatique.

Un système d'exploitation, un logiciel où les informations sont enregistrées sur un support informatique.

Un certain nombre de logiciels utilitaires sont généralement livrés avec les systèmes d'exploitation.

Cependant, ces programmes ne sont pas considérés comme faisant partie du système d'exploitation. Bien que les logiciels utilitaires livrés avec système d'exploitation soient plus complet et sophistiqué, les utilisateurs installent souvent des logiciels utilitaires TIERS en remplacement ou en complément de ceux fournis avec le système d'exploitation.

Types d'utilitaires :

-Utilitaires de sauvegarde : Ils enregistrent des copies des informations stockées sur un fichier, un dossier ou un disque et restaurent les fichiers sélectionnés (par exemple en cas de suppression accidentelle) ou le disque entier (par exemple en cas de panne de disque cassé).

-Utilitaires antivirus : Analysent et suppriment les logiciels malveillants.

-Utilitaires d'archivage de fichiers : ils combinent plusieurs fichiers en une seule archive ou une série d'archives pour faciliter le transport ou le stockage. Ceux-ci peuvent inclure des capacités de compression et de cryptage.

- etc.

Les logiciels utilitaires couvrent tous les domaines de l'informatique. Certains utilitaires sont dédiés à l'amélioration des fonctionnalités du système, tels que les programmes de presse-papiers, de capture d'écran ou de copie de fichiers. Ces fonctionnalités sont déjà disponibles dans le système d'exploitation, mais des composants tiers peuvent les remplacer, en ajoutant souvent des paramètres avancés.

*Pour maintenir les pilotes et les utilitaires de contrôle de version du système d'exploitation qui peuvent fournir des mises à jour de sécurité.

En bref, les utilitaires peuvent également être définis comme des logiciels qui effectuent des activités de gestion ou d'exploitation.

BIBLIOGRAPHIE

June Jamrich Parsons et Dan Oja, *New Perspectives on Computer Concepts 2014: Comprehensive*, Cengage Learning, 2013 ([lire en ligne](#) [\[archive\]](#)), p. 129

<https://www.ionos.fr/digitalguide/sites-internet/creation-de-sites-internet/expressions-regulieres/>

TABLE DE MATIERES

Expressions régulières	3
Fonctionnement d'une expression régulière	4
Les enjeux liés à l'emploi d'expression régulière	5
Utilitaires	8
Types d'utilitaires.....	9
BIBLIOGRAPHIE	10