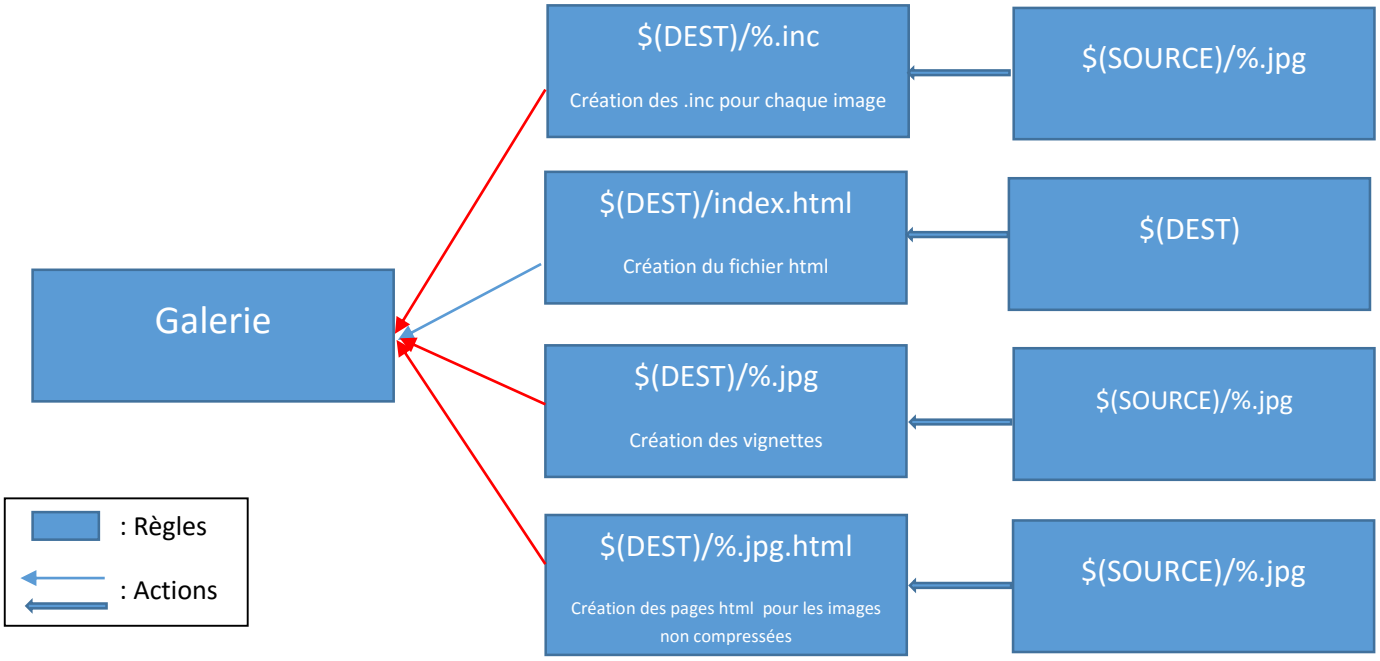
TP UNIX : GENERATEUR DE « GALERIE D'IMAGES» EN HTML

BOUTON Paul, BAYARD Guillaume

G2

I-Réponses aux questions du sujet

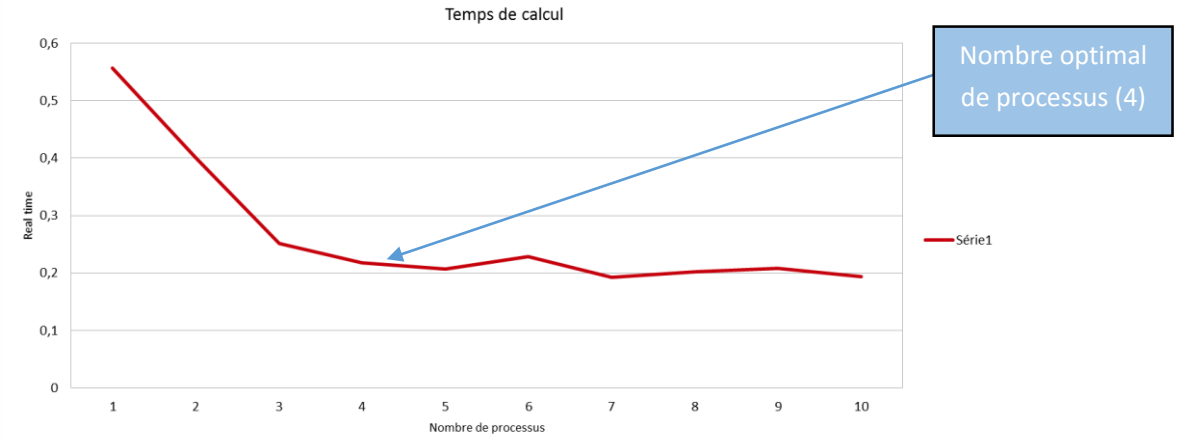
Graphe des makefiles



Les flèches rouges indiquent que les règles sont utilisées plusieurs fois (une fois par image).

Performances

1-Test makefile



Pour chaque nombre de processus, nous avons lancé dix tests sur 6 images et pris la moyenne (en secondes) des temps « real time » obtenus. On constate un gain de performance important jusqu'à 3 processus. Ensuite, la vitesse d'exécution du code ne semble plus évoluer. Ce ralentissement est dû au processeur du PC sur lequel ont été effectués ces tests (Core i3 M370). En effet, ce CPU possède deux cœurs physiques chacun divisés en deux cœurs logiques. Ainsi, il

est normal de voir la vitesse d'exécution multipliée par deux en passant de un à deux processus (on utilise les deux cœurs physiques), puis le gain de performances ralentit à partir de trois processus (utilisation des 4 cœurs logiques moins performants que 4 cœurs physiques réels).

2-Script

Nous avons également réalisé des tests de performance pour notre version « script ». Le temps moyen d'exécution de notre script est de 0.633 seconde (pour 10 tests sur 6 images effectués sur le même PC que pour la version makefile)) ce qui est supérieur au temps moyen d'exécution de la version makefile même avec un seul processus lancé. Ainsi, nous pouvons conclure que la version makefile est bien plus performante que la version script (200% plus performant si on exécute le code avec 4 processus, le nombre optimal).

II Points forts de notre code

Les options à base de *set –x* ou *set –v* n'étant pas envisageables, nous avons créé une fonction *mode_verb* dans le script utilities.sh. Ainsi, le mode verbe n'alourdit pas notre script *galerie_shell*. Effectivement, pour sélectionner le texte à afficher, nous fournissons à notre fonction une série de différents paramètres qui définissent la commande exécutée et donc le texte à afficher. Nous avons considéré que les étapes les plus importantes étaient la navigation d'un dossier vers un autre, la copie d'un fichier vers un répertoire, la conversion d'une image au format 200*200, l'éventuelle création d'une image, et enfin pour chaque image la création de sa vignette et de sa page HTML.
Chaque description de notre mode verbeux utilise des mots du langage courant et permet donc à un utilisateur qui n'a pas étudié le script de comprendre immédiatement quelles sont les commandes qui sont exécutées par ce dernier. Notre mode verbeux est donc compréhensible pour tout utilisateur tout en restant précis.
Nous avons également réussi à créer des pages HTML indépendantes pour chaque image non compressée, auxquelles on accède en cliquant sur la vignette correspondante de la page principale.
Enfin, nous avons géré un grand nombre d'exceptions en ce qui concerne les caractères spéciaux (grâce notamment à la commande sed). Ainsi, les caractères spéciaux du point de vue de l'URL ou du code HTML ont été pour la plupart traités (#, ? « »,<, >, ,…).

III Pistes d'améliorations

Premièrement, nous regrettons de ne pas avoir pu améliorer le design de notre galerie HTML. En effet, cette dernière reprend les codes visuels de celle présentée en exemple. De plus, nous ne sommes pas parvenus à mettre en place un système de navigation entre les pages contenant les images non compressées (image suivante, image précédente, retour à la galerie,…). Enfin, il reste quelques cas de caractères spéciaux non pris en charge (* suivi de plusieurs espaces, un espace seul,…).