

Spécification Technique de Besoin

Version	1.5
Date	11 mai 2012
Rédigé par	Zakaria ADDI
Relu par	Claire SMETS, Zineb ISSAAD

MISES À JOUR

Version	Date	Modifications réalisées
0.1	21/11/2011	Création
0.2	08/01/2012	Mise à jour
1.0	16/01/2012	Mise à jour
1.1	24/01/2012	Modification
1.2	03/02/2012	Modifications
1.3	09/02/2012	Modifications
1.4	11/02/2012	Modifications (ajout et suppression de cas d'utilisation)
1.5	14/02/2012	Modifications des tableaux de use case
2.1	23/04/2012	Révision des objectifs

Table des matières

1	Objet	4
2	Documents applicables et de référence	4
3	Terminologie et sigles utilisés	4
3.1	Présentation de la mission du produit logiciel	4
4	Exigences fonctionnelles	5
4.1	Formation de l'agrégat	5
4.2	Activation de l'agrégat	6
4.3	Lecture d'un bloc	7
4.4	Écriture d'un bloc	8
4.5	Choix de la stratégie de gestion de cache	8
4.6	Transfert du cache vers le segment de mémoire principal	10
4.7	"Flush" du cache	11
4.8	Exigences fonctionnelles détaillées	12
5	Exigences opérationnelles	13
6	Exigences d'interface	13
7	Exigences de qualité	13
8	Exigences de réalisation	13

1 Objet

besoin opérationnel : implanter un module noyau permettant de gérer deux segments de mémoire dans l'environnement Linux, l'un jouant le rôle de cache à l'autre ;

objectifs techniques :

- créer et intégrer le module noyau du système ;
- choix et implémentation d'un protocole de mapping ;

contraintes et recommandations :

- compétences techniques nouvelles et important travail de documentation nécessaires à la réalisation du pilote ;
- nécessité d'un SSD ;

résultat attendu : un système de cache entre deux segments de mémoire.

2 Documents applicables et de référence

- Projet1-2-3.pdf - Document de besoin initial fourni par le client.

3 Terminologie et sigles utilisés

Bloc : unité logique de données qui correspond à un ou plusieurs secteurs sur un disque.

Fichier : sous les systèmes d'exploitation avec un système de noyau linux, tout est "fichier" : processus, partitions, ...

HDD : Hard Drive Disk, périphérique de stockage de masse.

Il consomme plus et ses performances sont moindres qu'un SSD mais le prix d'acquisition est beaucoup moins élevé.

Module noyau : programme qui peut être chargé dynamiquement dans le noyau.

PBV : Périphérique Bloc Virtuel. Il s'agira ici d'un fichier spécial de type bloc représentant deux disques (SSD et HDD). Il est cependant possible qu'il ne soit composé que du HDD.

Pilote : programme gérant les opérations entre un périphérique et le reste du système.

Segment de mémoire : espace d'adressage indépendant défini par l'adresse où il commence et sa taille.

SGF : un Système de Gestion de Fichiers "est une façon de stocker les informations et de les organiser dans des fichiers". (Wikipedia)

SSD : Solid State Disk, périphérique de stockage qui utilise de la mémoire flash.

Les performances sont élevées et les consommations d'énergie basses mais le coût par Go est élevé.

Swap : zone mémoire réservée sur l'espace de stockage de masse et/ou sur le PBV servant de mémoire d'appoint à la mémoire principale.

3.1 Présentation de la mission du produit logiciel

Le produit aura pour mission de gérer simultanément deux segments de mémoire, l'un servant de cache au second.

Deux modes seront implantés : le mode sécurité et le mode économie. Lorsque le mode sécurité sera activé, l'écriture se fera sur les deux segments "en même temps", tandis qu'avec le mode économie, l'écriture sur le segment de mémoire principal ne se fera que lorsque l'information sera vidée du cache.

Il sera possible de paramétrer l'agrégat : mode souhaité, segments présents dans l'agrégat, possibilité de le désactiver, ...

Optionnellement, les deux segments de mémoire seront remplacés par un SSD et un HDD.

4 Exigences fonctionnelles

Reference	Fonctionnalité Globale	Acteurs	Priorité
F-Gl-10	Formation de l'agrégat	Utilisateur	Indispensable
F-Gl-20	Activation de l'agrégat	Utilisateur	Indispensable
F-Gl-30	Lecture de bloc	Utilisateur	Indispensable
F-Gl-40	Ecriture de bloc	Processus client	Indispensable
F-Gl-50	Choix de la stratégie de gestion du cache	Utilisateur	Important
F-Gl-60	Transfert du cache vers le segment de mémoire principal	Utilisateur et pilote	Important
F-Gl-70	"Flush" du cache	Utilisateur et pilote	Important

4.1 Formation de l'agrégat

Nom: C1		Formation de l'agrégat	
Acteurs concernés		Pilote, utilitaire et utilisateur	
Description		Les deux segments de mémoire du futur agrégat sont définis par défaut.	
Préconditions		Les deux segments de mémoire sont reconnus par le noyau.	
Evénements déclenchants		L'utilisateur demande à changer les segments de mémoire de l'agrégat.	
Conditions d'arrêt		Les segments de mémoire du futur agrégat seront enregistrés par défaut. À chaque démarrage ceux-ci formeront le nouvel agrégat.	
Description du flot d'événements principal:			
Acteurs:			
Utilisateur		Utilitaire	
1. demande de formation de l'agrégat ; 2. choix des segments de mémoire.			
Flots secondaires:			
Flots d'exceptions:		<ul style="list-style-type: none">• segment de mémoire introuvable (chemin ou nom incorrect) : demande un autre segment ;• segment de mémoire invalide (lecteur de disquette, souris, ...) : demande un autre segment ;• si la taille du cache est supérieure à celle du du segment de mémoire principal : d'autres segments de mémoire à indiquer.	

4.2 Activation de l'agrégat

Elle se fera automatiquement lors du démarrage de la machine si les segments de mémoire spécifiés sont corrects (cf les flots secondaire et d'exception ci-dessous). Cependant, dans ce cas précis, l'action se faisant sans demande d'avis à l'utilisateur, il n'est pas détaillé.

Nom: C2	Activation de l'agrégat à la demande de l'utilisateur
Acteurs concernés	OS, pilote
Description	Les segments de mémoire ne forment désormais plus qu'un et ne sont plus dissociables du point de vue de l'utilisateur.
Préconditions	Les deux segments de mémoire ont été spécifiés dans la formation de l'agrégat et les chemins sont valides.
Événements déclenchants	L'utilisateur demande à activer l'agrégat.
Conditions d'arrêt	Le cache et le segment de mémoire principal ne forment plus qu'un agrégat, prêt à être utilisé. L'utilisateur ne peut plus les différencier.
Description du flot d'événements principal:	
Acteurs:	
Utilisateur	Utilitaire
1. demande d'activation de l'agrégat ;	2. récupère les chemins des segments de mémoire à regrouper ; 3. active l'agrégat.
Flots secondaires:	
Flots d'exceptions:	<ul style="list-style-type: none">• segment de mémoire déjà utilisé : prévenir l'utilisateur ;• l'uuid d'un des segments de mémoire indiqués est différent de celui présent dans le fichier de configuration : prévenir l'utilisateur et lui demander de reformer l'agrégat.

4.3 Lecture d'un bloc

Pour la lecture et l'écriture d'un fichier, deux modes seront implantés :

- le mode économie qui privilégiera les économies d'énergie : les accès sur le segment de mémoire principal ne seront pas effectués systématiquement.
- le mode sécurité qui écrira les données sur le cache et le segment de mémoire principal : permet d'avoir les données dupliquées et d'être sûr qu'aucune n'est perdue.

Nom: C3	Lecture d'un bloc en mode économie ou sécurité	
Acteurs concernés	Pilote, OS et processus client	
Description	L'utilisateur veut lire des données sur l'agrégat.	
Préconditions	L'agrégat est activé et les numéros de blocs demandés sont valides.	
Evénements déclenchants	Demande de lecture de l'utilisateur sur un bloc.	
Conditions d'arrêt	Les données ont été lues et transmises.	
Description du flot d'événements principal:		
Acteurs:		
Utilisateur	Pilote	
1. demande de lecture d'un bloc ;	2.1. si les données sont sur le cache alors 1. envoie la demande de lecture au segment de mémoire principal ; 2. récupère les données ; 3. stocke les données sur le cache ; 2.2. si les données sont sur le cache alors 1. envoie la demande de lecture au cache ; 2. récupère les données ; 3. transmet les données ;	
4. (peut manipuler les données.)		
Flots secondaires:		
Flots d'exceptions:		

4.4 Écriture d'un bloc

Lors de l'écriture sur le cache, on suppose qu'il y a assez de place dessus : une stratégie de gestion du cache sera implantée mais ne sera pas appelée lors de l'écriture : cela ferait perdre trop de temps (et serait très inconfortable pour l'utilisateur) si lors de l'écriture d'un fichier l'on devait déplacer plusieurs méga voir giga-octets du cache vers le segment de mémoire principal. Le transfert se fera donc dès qu'un seuil critique sera passé, et la taille des données transférées sera assez importante pour que la manipulation ne soit pas trop fréquente. La tâche sera effectuée en tâche de fond.

Nom: C4	Écriture d'un bloc en mode économie
Acteurs concernés	Pilote, OS et processus client
Description	L'utilisateur veut écrire dans un bloc.
Préconditions	L'agrégat est activé en mode économie et les blocs à écrire ont des numéros valides.
Événements déclenchants	Demande d'écriture de l'utilisateur dans un bloc.
Conditions d'arrêt	Les données ont été écrites sur l'agrégat.
Description du flot d'événements principal:	
Acteurs:	
Utilisateur	Pilote
1. demande l'écriture d'un bloc ;	2. écrire les blocs sur le cache.
Flots secondaires:	
Flots d'exceptions:	<ul style="list-style-type: none"> Erreur d'écriture sur le cache : prévenir l'utilisateur

Nom: C5	Écriture d'un bloc en mode sécurité
Acteurs concernés	Pilote, OS et processus client
Description	L'utilisateur veut écrire dans un bloc.
Préconditions	L'agrégat est activé en mode sécurité et les blocs à écrire ont des numéros valides.
Événements déclenchants	Demande d'écriture de l'utilisateur dans un bloc.
Conditions d'arrêt	Les données ont été écrites sur l'agrégat.
Description du flot d'événements principal:	
Acteurs:	
Utilisateur	Pilote
1. demande l'écriture d'un bloc ;	2. écrire les blocs sur le cache ; 3. écrire les blocs sur le segment de mémoire principal.
Flots secondaires:	
Flots d'exceptions:	<ul style="list-style-type: none"> Erreur d'écriture sur l'un des deux segments de mémoire : prévenir l'utilisateur

4.5 Choix de la stratégie de gestion de cache

Ici, on sera face à deux cas :

- le pilote est en cours d'utilisation, il faudra s'assurer que toutes les opérations en cours se terminent et que l'intégrité du PBV soit préservée. L'utilisateur aura peut-être à redémarrer

son poste de travail.

- sinon, on pourra effectuer directement le changement, qui deviendra effectif lors de la prochaine activation.

activation.	
Nom: C6	Choix de la stratégie de gestion du cache
Acteurs concernés	Utilitaire, utilisateur et pilote
Description	L'utilisateur veut choisir la stratégie de gestion de cache.
Préconditions	L'agrégat est formé.
Evénements déclenchants	Demande de changement de la stratégie de gestion de cache.
Conditions d'arrêt	Le changement est effectué et les modifications ont bien été réalisées.
Description du flot d'événements principal:	
Acteurs:	
Utilisateur	Utilitaire
1. demande de changement de stratégie ; 2. choix de la stratégie ; 4. l'utilisateur est informé de la stratégie mise en place et l'opération est terminée.	3.1 si la stratégie demandée est effectivement différente, elle est changée ; 3.2 sinon rien n'est fait ;
Flots secondaires:	
Flots d'exceptions:	

4.6 Transfert du cache vers le segment de mémoire principal

Nom: C7		Transfert du cache vers la mémoire principale	
Acteurs concernés		Utilisateur et pilote	
Description		L'utilisateur souhaite dupliquer des blocs du cache vers le segment de mémoire principal.	
Préconditions		Les blocs du cache sont écrits sur le segment de mémoire principale s'ils n'y étaient pas déjà.	
Evénements déclenchants		L'utilisateur demande de synchroniser le cache et le segment de mémoire principale.	
Conditions d'arrêt		L'utilisateur ferme l'utilitaire, et tous les blocs sont dupliqués.	
Description du flot d'événements principal:			
Acteurs:			
Utilisateur		Utilitaire	
1. demande de synchronisation du cache ;		2. pour chaque bloc de données présent sur le cache faire : <ul style="list-style-type: none">○ s'il est présent (dans le même état) sur le segment de mémoire principal alors : ne rien faire ;○ s'il n'est pas présent sur le segment de mémoire principal ou qu'il a été modifié depuis : l'écrire ou le réécrire (à la place) sur le segment de mémoire principal.	
Flots secondaires:			
Flots d'exceptions:		<ul style="list-style-type: none">• certains blocs ne peuvent être transférés (secteurs défaillants) : les autres blocs sont tout de même transférés mais l'utilisateur est prévenu de l'erreur.	

4.7 "Flush" du cache

Nom: C8	"Flush" du cache
Acteurs concernés	Utilisateur et pilote
Description	L'utilisateur souhaite vider le cache.
Préconditions	Les blocs du cache sont écrits sur le segment de mémoire principal s'ils n'y étaient pas déjà, puis l'intégralité du contenu du cache est effacé.
Evénements déclenchants	L'utilisateur demande de vider le cache.
Conditions d'arrêt	L'utilisateur ferme l'utilitaire, le cache est vide mais aucune information n'est perdue : elles sont sur le segment de mémoire principal.
Description du flot d'événements principal:	
Acteurs:	
Utilisateur	Utilitaire
1. demande de "Flush" du cache	2. pour chaque bloc de données présent sur le cache faire : <ul style="list-style-type: none">o s'il est présent (dans le même état) sur le segment de mémoire principal alors : ne rien faire ;o s'il n'est pas présent sur le segment de mémoire principal ou qu'il a été modifié depuis : l'écrire ou le réécrire (à la place) sur le segment de mémoire principal. 3. vider le cache.
Flots secondaires:	
Flots d'exceptions:	<ul style="list-style-type: none">certains blocs ne peuvent être transférés (secteurs défaillants) : les autres blocs sont tout de même transférés puis supprimés du cache, mais l'utilisateur est prévenu que le cache n'est pas complètement vide et que certaines données ne sont pas présentes sur le segment de mémoire principal.

4.8 Exigences fonctionnelles détaillées

Reference	Fonctionnalité	Priorité
F-FN-10	Formation de l'agrégat : si les segments de mémoire sélectionnés n'existent pas ou sont mal spécifiés rien ne se passe.	Indispensable
F-FN-20	Formation de l'agrégat : si les segments de mémoire sélectionnés ne sont pas du bon type rien ne se passe.	Indispensable
F-FN-30	Formation de l'agrégat : si la taille du cache est supérieure à celle du segment principal, on redemande d'autres segments de mémoire.	Indispensable
F-FN-40	Formation de l'agrégat : si tous les segments de mémoire sont détectés, l'agrégat est effectivement formé sinon on demande un autre périphérique.	Indispensable
F-FN-50	Activation de l'agrégat : si le segment de mémoire est déjà utilisé, on prévient l'utilisateur.	Indispensable
F-FN-60	Activation de l'agrégat : si les chemins corrects et que les segments de mémoire ne sont pas utilisés par d'autres processus, activation de l'agrégat.	Indispensable
F-FN-70	Lecture d'un bloc en mode économies et sécurité : si les droits d'utilisateurs sont suffisants, les données sont lues, sinon rien ne se passe.	Indispensable
F-FN-80	Écriture d'un bloc en mode économie ou sécurité : si l'utilisateur ne dispose pas des droits, rien ne se passe.	Indispensable
F-FN-90	Écriture d'un bloc en mode économie : écriture des blocs sur le cache.	Indispensable
F-FN-100	Écriture d'un bloc en mode sécurité : écriture des blocs sur le cache et le segment de mémoire principal.	Indispensable
F-FN-110	Transfert du cache vers le segment de mémoire principal : si l'espace libre est insuffisant, aucun transfert n'est effectué. Sinon tous les blocs du cache (non présents sur le HDD) sont copiés sur le segment principal.	Important
F-FN-120	"Flush" du cache reprend le cas précédent mais en supprimant tous les blocs du cache.	Important

5 Exigences opérationnelles

Reference	Fonctionnalité	Priorité
F-FO-10	Le PBV sera l'agrégat de deux segments de mémoire	Indispensable
F-FO-20	Support des opérations usuelles sur les périphériques de stockage de masse	Indispensable
F-FO-30	Le temps d'accès sera optimisé	Optionnel

6 Exigences d'interface

Reference	Fonctionnalité	Priorité
F-FI-10	L'interface du pilote devra respecter les normes du standard POSIX.	Indispensable

7 Exigences de qualité

Pas d'exigence de qualité.

8 Exigences de réalisation

Reference	Fonctionnalité	Priorité
F-FR-10	Un périphérique bloc virtuel qui représentera l'agglomérat de deux segments de mémoire, l'un étant un cache et l'autre représentant la mémoire principale de l'agrégat	Indispensable
F-FR-20	Un pilote qui puisse communiquer avec le Périphérique Bloc Virtuel	Indispensable