

Cours Système

D.Revuz

17 février 2005

Table des matières

1	Introduction	1
1.1	Unix	1
1.1.1	Pourquoi unix?	1
1.1.2	le succès d'Unix et de linux	1
1.1.3	Des points forts	1
1.1.4	Des points faibles	2
1.2	Structure générale des systèmes d'exploitation	2
1.2.1	Les couches fonctionnelles	3
1.2.2	L'architecture du système	3
1.2.3	L'architecture du noyau	5
1.3	historique	5
2	Système de Gestion de Fichiers	7
2.1	Le concept de fichier	7
2.2	Fichiers ordinaires / Fichiers spéciaux.	8
2.3	Organisation utilisateur des Disques	8
2.4	Les inodes	9
2.5	Organisation des disques System V	10
2.6	Adressage des blocs dans les inodes	11
2.7	Allocation des inodes d'un disque	11
2.8	Allocation des blocs-disque	11
3	Le Buffer Cache	17
3.1	Introduction au buffer cache	17
3.1.1	Avantages et désavantages du buffer cache	17
3.2	Le buffer cache, structures de données.	17
3.2.1	La liste doublement chaînée des blocs libres	18
3.3	L'algorithme de la primitive <code>getblk</code>	18
4	La bibliothèque standard	23
4.1	Les descripteurs de fichiers.	23
4.1.1	Ouverture d'un fichier	24
4.1.2	Redirection d'un descripteur : <code>freopen</code>	24
4.1.3	Création de fichiers temporaires	25
4.1.4	Ecriture non formatée	25
4.1.5	Accès séquentiel	26
4.1.6	Manipulation du pointeur de fichier	26
4.1.7	Un exemple d'accès direct sur un fichier d'entiers.	26
4.1.8	Les autres fonctions de déplacement du pointeur de fichier.	26
4.2	Les tampons de fichiers de <code>stdlib</code> .	27
4.2.1	Les modes de bufferisation par défaut.	27
4.2.2	Manipulation des tampons de la bibliothèque standard.	27

4.3	Manipulation des liens d'un fichier	29
4.4	Lancement d'une commande shell	29
4.5	Terminaison d'un processus	30
4.6	Gestion des erreurs	31
4.7	Création et destruction de répertoires	31
5	Appels système du Système de Gestion de Fichier	33
5.1	<code>open</code>	33
5.1.1	Déroulement interne d'un appel de <code>open</code>	35
5.2	<code>creat</code>	36
5.3	<code>read</code>	36
5.4	<code>write</code>	36
5.5	<code>lseek</code>	37
5.6	<code>dup</code> et <code>dup2</code>	37
5.7	<code>close</code>	38
5.8	<code>fcntl</code>	38
6	Les processus	41
6.1	Introduction aux processus	41
6.1.1	Création d'un processus - <code>fork()</code>	41
6.2	Format d'un fichier exécutable	42
6.3	Chargement/changement d'un exécutable	42
6.4	zone <code>u</code> et table des processus	43
6.5	<code>fork</code> et <code>exec</code> (revisités)	43
6.6	Le contexte d'un processus	45
6.7	Commutation de mot d'état et interruptions.	45
6.8	Les interruptions	46
6.9	Le problème des cascades d'interruptions	47
6.9.1	Etats et transitions d'un processus	47
6.9.2	Listes des états d'un processus	47
6.10	Lecture du diagramme d'état.	48
6.11	Un exemple d'exécution	49
6.12	La table des processus	49
6.13	La zone <code>u</code>	50
6.14	Accès aux structures <code>proc</code> et <code>user</code> du processus courant	50
6.14.1	Les informations temporelles.	50
6.14.2	Changement du répertoire racine pour un processus.	51
6.14.3	Récupération du PID d'un processus	51
6.14.4	Positionnement de l'euid, ruid et suid	51
6.15	Tailles limites d'un processus	52
6.15.1	Manipulation de la taille d'un processus.	52
6.15.2	Manipulation de la valeur <code>nice</code>	52
6.15.3	Manipulation de la valeur <code>umask</code>	52
6.16	L'appel système <code>fork</code>	53
6.17	L'appel système <code>exec</code>	53
7	L'ordonnancement des processus	55
7.1	Le partage de l'unité centrale	55
7.1.1	Famine	56
7.1.2	Stratégie globale	56
7.1.3	Critères de performance	57
7.2	Ordonnancement sans préemption.	57
7.3	Les algorithmes préemptifs	57
7.3.1	Round Robin (tourniquet)	58

7.3.2	Les algorithmes à queues multiples	58
7.4	Multi-level-feedback round robin Queues	58
7.4.1	Les niveaux de priorité	58
7.4.2	Evolution de la priorité	59
7.4.3	Les classes de priorité	60
8	La mémoire	61
8.0.4	les mémoires	61
8.0.5	La mémoire centrale	61
8.1	Allocation contiguë	63
8.1.1	Pas de gestion de la mémoire	63
8.1.2	Le moniteur résidant	63
8.1.3	Le registre barrière	63
8.1.4	Le registre base	63
8.1.5	Le swap	65
8.1.6	Le coût du swap	65
8.1.7	Utilisation de la taille des processus	65
8.1.8	Swap et exécutions concurrentes	66
8.1.9	Contraintes	66
8.1.10	Deux solutions existent	66
8.1.11	Les problèmes de protection	66
8.1.12	Les registres doubles	66
8.2	Ordonnancement en mémoire des processus	67
8.3	Allocation non-contiguë	68
8.3.1	Les pages et la pagination	68
8.3.2	Ordonnancement des processus dans une mémoire paginée	68
8.3.3	Comment protéger la mémoire paginée	69
8.3.4	La mémoire segmentée	69
9	La mémoire virtuelle	71
9.0.5	Les overlays	71
9.0.6	Le chargement dynamique	72
9.1	Demand Paging	72
9.1.1	Efficacité	73
9.2	Les algorithmes de remplacement de page	75
9.2.1	Le remplacement optimal	75
9.2.2	Le remplacement peps (FIFO)	75
9.2.3	Moins récemment utilisée LRU	76
9.2.4	L'algorithme de la deuxième chance	76
9.2.5	Plus fréquemment utilisé MFU	76
9.2.6	Le bit de saleté (Dirty Bit)	77
9.3	Allocation de pages aux processus	77
9.4	L'appel fork et la mémoire virtuelle	78
9.5	Projection de fichiers en mémoire	78
9.6	Les conseils et politiques de chargement des zones mmappées	80
9.7	Chargement dynamique	81
10	Tubes et Tubes Nommés	83
10.1	Les tubes ordinaires (<i>pipe</i>)	83
10.2	Création de tubes ordinaires	83
10.3	Lecture dans un tube	85
10.4	Ecriture dans un tube	86
10.5	Interblocage avec des tubes	86
10.6	Les tubes nommés	86

10.6.1	Ouverture et synchronisation des ouvertures de tubes nommés	87
10.6.2	Suppression d'un tube nommé	87
10.6.3	les appels <code>popen</code> et <code>pclose</code>	87
11	Les signaux	89
11.0.4	Provenance des signaux	89
11.0.5	Gestion interne des signaux	89
11.0.6	L'envoi de signaux : la primitive <code>kill</code>	90
11.1	La gestion simplifiée avec la fonction <code>signal</code>	91
11.1.1	Un exemple	91
11.2	Problèmes de la gestion de signaux ATT	91
11.2.1	Le signal <code>SIGCHLD</code>	93
11.3	Manipulation de la pile d'exécution	94
11.4	Quelques exemples d'utilisation	95
11.4.1	L'appel <code>pause</code>	95
11.5	La norme POSIX	96
11.5.1	Le blocage des signaux	96
11.5.2	<code>sigaction</code>	97
11.5.3	L'attente d'un signal	97
12	Les verrous de fichiers	99
12.1	Caractéristiques d'un verrou	99
12.2	Le mode opératoire des verrous	99
12.3	Manipulation des verrous	100
12.4	Utilisation de <code>fcntl</code> pour manipuler les verrous	101
13	Algorithmes Distribués & Interblocages	103
13.1	exemples	103
13.1.1	Les méfaits des accès concurrents	103
13.1.2	Exclusion mutuelle	104
13.2	Mode d'utilisation des ressources par un processus	105
13.3	Définition de l'interblocage (deadlock)	105
13.4	Quatre conditions nécessaires à l'interblocage	105
13.5	Les graphes d'allocation de ressources	105
14	Sécurité et Sûreté de fonctionnement	107
14.1	Protection des systèmes d'exploitation	107
14.2	Généralités sur le contrôle d'accès	108
14.2.1	Domaines de protection et matrices d'accès	109
14.2.2	Domaines de protection restreints	109
14.2.3	Avantages des domaines de protections restreints	110
14.3	Le cheval de Troie	110
14.4	Le confinement	110
14.5	les mécanismes de contrôle	110
14.5.1	Application des capacités au domaines de protection restreints	112
14.6	Les ACL	115
14.6.1	Appels systèmes <code>setacl</code> et <code>getacl</code>	115
14.6.2	Autres pistes sur la sécurité	116
15	Multiplexer des entrées-sorties	119
15.1	Gerer plusieurs canaux d'entrée sortie	119
15.1.1	Solution avec le mode non bloquant	119
15.1.2	Utiliser les mécanismes asynchrones	119
15.2	Les outils de sélection	119

15.2.1	La primitive <code>select</code>	119
15.2.2	La primitive <code>poll</code>	121
15.2.3	Le périphérique <code>poll</code>	122
15.2.4	Les extensions de <code>read</code> et <code>write</code>	123
15.3	une solution multi-activités	123
16	Les threads POSIX	125
16.0.1	Description	125
16.0.2	<code>fork</code> et <code>exec</code>	125
16.0.3	<code>clone</code>	127
16.0.4	Les noms de fonctions	127
16.0.5	les noms de types	127
16.0.6	Attributs d'une activité	128
16.0.7	Création et terminaison des activités	128
16.1	Synchronisation	128
16.1.1	Le modèle <code>fork/join</code> (Paterson)	129
16.1.2	Le problème de l'exclusion mutuelle sur les variables gérées par le noyau	129
16.1.3	Les sémaphores d'exclusion mutuelle	129
16.1.4	Utilisation des sémaphores	130
16.1.5	Les conditions (événements)	130
16.2	Ordonnancement des activités	132
16.2.1	L'ordonnancement POSIX des activités	132
16.3	Les variables spécifiques à une thread	133
16.3.1	Principe général des données spécifiques, POSIX	134
16.3.2	Création de clés	134
16.3.3	Lecture/écriture d'une variable spécifique	134
16.4	Les fonctions standards utilisant des zones statiques	134
17	Clustering	135
17.1	Le clustering sous linux	135
18	Bibliographie	137
18.1	Webographie	137

Cours de conception de systèmes et d'utilisation d'UNIX

Ce poly est à l'usage des étudiants de la filière Informatique et Réseaux de l'école d'ingénieurs Ingénieurs 2000 UMLV et de la troisième année de licence d'informatique de Marne la Vallée comme support du cours SYSTÈMES d'EXPLOITATION.

Cette version du , apporte de nombreuses corrections de typo et autre, je remercie David Lecorfec pour sa lecture attentive, et les remarques sur le fond seront prises en compte dans la prochaine version.

Ce poly a une version HTML disponible sur le Web à l'adresse suivante :

<http://www-igm.univ-mlv.fr/~dr/NCS/>

Ce document a de nombreux défauts en particulier son manque d'homogénéité, et une absence d'explications dans certaines parties (explication données en général oralement en cours).

Au menu l'essentiel d'UNIX : SGF, processus, signaux, mémoire, mémoire virtuelle, manipulation des terminaux, tubes, IPC. Quelques détours : micro-noyaux, sécurité. Un chapitre important mais un peu court : les problèmes de programmation distribué et des interblocages (améliorations en cours fin 2004).

Prérequis : pour la partie conceptuelle des notions de programmation et d'algorithmique sont nécessaires pour profiter pleinement du cours, pour la partie technique une compétence raisonnable en C est nécessaire, en particulier les notions de pointeurs d'allocation dynamique doivent être maîtrisées, les 4 méthodes d'allocation principale du C doivent être maîtrisées ! (text,static,auto,heap).

Évolutions futures : dr@univ-mlv.fr (j'attends vos remarques), uniformisation de la présentation, nettoyage des points obscurs, corrections orthographiques, complément sur fcntl, ioctl, plus d'exemples, des sujets de projets , des sujets d'examen.