

Содержание

Список примеров	12
1. Для нетерпеливого читателя	16
1.1. Последовательная программа	17
1.2. Ускорение работы за счет параллелизма	19
1.3. Параллельная программа, использующая процессы .	19
1.4. Параллельная программа, использующая потоки . .	25
1.5. Параллельная программа, использующая MPI	29
2. Пути повышения производительности процессоров	32
2.1. CISC- и RISC-процессоры	32
2.2. Основные черты RISC-архитектуры	33
2.3. Конвейеризация	34
2.4. Кэш-память	41
2.5. Многопроцессорные архитектуры	47
2.5.1. Основные архитектуры	47
2.5.2. Комбинированные архитектуры	48
2.5.3. Обанкротившиеся архитектуры	50
2.6. Поддержка многозадачности и многопроцессорности	51
2.7. Использование параллелизма процессора	53
3. Пути повышения производительности оперативной памяти	69
4. Организация данных во внешней памяти	71
5. Основные положения	72
5.1. Основные определения	72
5.2. Виды ресурсов	78
5.3. Типы взаимодействия процессов	80
5.4. Состояния процесса	84

6. Стандарты на операционные системы UNIX	86
6.1. Стандарт BSD 4.3	86
6.2. Стандарт UNIX System V Release 4	87
6.3. Стандарт POSIX 1003	87
6.4. Стандарт UNIX X/Open	87
7. Управление процессами	88
7.1. Функция fork	88
7.2. Функции execl, execv	91
7.3. Функция waitpid	91
7.4. Функция kill	95
7.5. Функция signal	95
8. Синхронизация и взаимодействие процессов	102
8.1. Разделяемая память	103
8.1.1. Функция shmget	105
8.1.2. Функция shmat	105
8.1.3. Функция shmdt	106
8.1.4. Функция shmctl	107
8.2. Семафоры	107
8.2.1. Функция semget	110
8.2.2. Функция semop	111
8.2.3. Функция semctl	111
8.2.4. Пример использования семафоров и разделяемой памяти	112
8.3. События	128
8.4. Очереди сообщений (почтовые ящики)	131
8.4.1. Функция msgget	133
8.4.2. Функция msgsnd	134
8.4.3. Функция msgrcv	134
8.4.4. Функция msgctl	135
8.4.5. Пример использования очередей	136
8.4.6. Функция pipe	143
8.5. Пример вычисления произведения матрицы на вектор	145

9. Управление потоками (threads)	167
9.1. Функция <code>pthread_create</code>	167
9.2. Функция <code>pthread_join</code>	168
9.3. Функция <code>sched_yield</code>	169
9.4. Пример простейшей multithread-программы	169
10. Синхронизация и взаимодействие потоков	171
10.1. Объекты синхронизации типа <code>mutex</code>	171
10.1.1. Функция <code>pthread_mutex_init</code>	174
10.1.2. Функция <code>pthread_mutex_lock</code>	175
10.1.3. Функция <code>pthread_mutex_trylock</code>	175
10.1.4. Функция <code>pthread_mutex_unlock</code>	176
10.1.5. Функция <code>pthread_mutex_destroy</code>	176
10.1.6. Пример использования <code>mutex</code>	176
10.2. Пример multithread-программы, вычисляющей интеграл	182
10.3. Объекты синхронизации типа <code>condvar</code>	182
10.3.1. Функция <code>pthread_cond_init</code>	184
10.3.2. Функция <code>pthread_cond_signal</code>	185
10.3.3. Функция <code>pthread_cond_broadcast</code>	185
10.3.4. Функция <code>pthread_cond_wait</code>	186
10.3.5. Функция <code>pthread_cond_destroy</code>	186
10.3.6. Пример использования <code>condvar</code>	186
10.4. Объекты синхронизации типа <code>barrier</code>	193
10.4.1. Функция <code>pthread_barrier_init</code>	194
10.4.2. Функция <code>pthread_barrier_wait</code>	194
10.4.3. Функция <code>pthread_barrier_destroy</code>	195
10.5. Пример вычисления произведения матрицы на вектор	195
10.6. Дисциплина доступа к памяти и эффективность . .	211
10.7. Решение задачи Дирихле для уравнения Пуассона .	222
11. Message Passing Interface (MPI)	253
11.1. Общее устройство MPI-программы	254
11.2. Сообщения	255
11.3. Коммуникаторы	258