

차례

역자 서문	i
모든 독자에게	iii
교강사에게	v
학생들에게	vi
감사의 글	vii
맺음말	x
참고 문헌	xi
그림 차례	xiii
제 1 장 이 책에 관한 대화	1
제 2 장 운영체제 개요	3
2.1 CPU 가상화	4
2.2 메모리 가상화	6
2.3 병행성	8
2.4 영속성	10
2.5 설계 목표	12
2.6 역사 약간	13
2.7 요약	16
참고 문헌	18
제 I 편 가상화	21
제 3 장 가상화에 관한 대화	23
제 4 장 프로세스의 개념	25
4.1 프로세스의 개념	26
4.2 프로세스 API	27
4.3 프로세스 생성: 좀 더 자세하게	27

4.4	프로세스 상태	29
4.5	자료 구조	31
4.6	요약	33
	참고 문헌	34
	속제	35
	문제	35
제 5장	막간: 프로세스 API	37
5.1	<code>fork()</code> 시스템 콜	37
5.2	<code>wait()</code> 시스템 콜	39
5.3	드디어, <code>exec()</code> 시스템 콜	40
5.4	왜, 이런 API를?	41
5.5	여타 API들	43
5.6	요약	44
	참고 문헌	45
	속제(코드)	46
	문제	46
제 6장	제한적 직접 실행 원리	49
6.1	기본 원리: 제한적 직접 실행	49
6.2	문제점 1: 제한된 연산	50
6.3	문제점 2: 프로세스 간 전환	54
6.4	병행성이 걱정	58
6.5	요약	59
	참고 문헌	61
제 7장	스케줄링: 개요	65
7.1	워크로드에 대한 가정	65
7.2	스케줄링 평가 항목	66
7.3	선입선출	66
7.4	최단 작업 우선	68
7.5	최소 잔여시간 우선	69
7.6	새로운 평가 기준: 응답 시간	70
7.7	라운드 로빈	71
7.8	입출력 연산의 고려	73
7.9	만병통치약은 없다(No More Oracle)	74
7.10	요약	74
	참고 문헌	76
	속제	77
	문제	77

제 8 장 스케줄링: 멀티 레벨 피드백 큐	79
8.1 MLFQ: 기본 규칙	79
8.2 시도 1: 우선순위의 변경	80
8.3 시도 2: 우선순위의 상향 조정	84
8.4 시도 3: 더 나은 시간 측정	84
8.5 MLFQ 조정과 다른 쟁점들	85
8.6 MLFQ: 요약	87
참고 문헌	88
숙제	90
문제	90
제 9 장 스케줄링: 비례 배분	91
9.1 기본 개념: 추첨권이 당신의 몫을 나타낸다	91
9.2 추첨 기법	92
9.3 구현	93
9.4 예제	94
9.5 추첨권 배분 방식	95
9.6 왜 결정론적(Deterministic) 방법을 사용하지 않는가	96
9.7 요약	97
참고 문헌	98
숙제	99
문제	99
제 10 장 멀티프로세서 스케줄링 (고급)	101
10.1 배경: 멀티프로세서 구조	101
10.2 동기화를 잊지 마시오	104
10.3 마지막 문제점: 캐시 친화성	105
10.4 단일 큐 스케줄링	105
10.5 멀티 큐 스케줄링	106
10.6 Linux 멀티프로세서 스케줄러	110
10.7 요약	110
참고 문헌	111
제 11 장 CPU 가상화에 관한 마무리 대화	113
제 12 장 메모리 가상화에 관한 대화	115
제 13 장 주소 공간의 개념	117
13.1 초기 시스템	117
13.2 멀티프로그래밍과 시분할	118

13.3	주소 공간	119
13.4	목표	120
13.5	요약	122
	참고 문헌	124
제 14장	막간: 메모리 관리 API	127
14.1	메모리 공간의 종류	127
14.2	<code>malloc()</code> 함수	128
14.3	<code>free()</code> 함수	130
14.4	흔한 오류	130
14.5	운영체제의 지원	134
14.6	기타 함수들	134
14.7	요약	134
	참고 문헌	135
	숙제(코드)	136
	문제	136
제 15장	주소 변환의 원리	139
15.1	가정	140
15.2	사례	140
15.3	동적(하드웨어-기반) 재배치	142
15.4	하드웨어 지원: 요약	145
15.5	운영체제 이슈	147
15.6	요약	148
	참고 문헌	151
	숙제	152
	문제	152
제 16장	세그멘테이션	153
16.1	세그멘테이션: 베이스/바운드(base/bound)의 일반화	153
16.2	세그먼트 종류의 파악	156
16.3	스택	157
16.4	공유 지원	158
16.5	소단위 대 대단위 세그멘테이션	159
16.6	운영체제의 지원	159
16.7	요약	161
	참고 문헌	162
	숙제	164
	문제	164

제 17 장	빈 공간 관리	165
17.1	가정	166
17.2	저수준 기법들	166
17.3	기본 전략	175
17.4	다른 접근법	177
17.5	요약	179
	참고 문헌	180
	숙제	181
	문제	181
제 18 장	페이징: 개요	183
18.1	간단한 예제 및 개요	183
18.2	페이지 테이블은 어디에 저장되는가	187
18.3	페이지 테이블에는 실제 무엇이 있는가	187
18.4	페이징: 너무 느림	189
18.5	메모리 트레이스	190
18.6	요약	193
	참고 문헌	194
	숙제	195
	문제	195
제 19 장	페이징: 더 빠른 변환 (TLB)	197
19.1	TLB의 기본 알고리즘	198
19.2	예제: 배열 접근	199
19.3	TLB 미스는 누가 처리할까	201
19.4	TLB의 구성: 무엇이 있나?	203
19.5	TLB의 문제: 문맥 교환	203
19.6	이슈: 교체 정책	207
19.7	실제 TLB	207
19.8	요약	208
	참고 문헌	210
	숙제 (측정)	212
	문제	212
제 20 장	페이징: 더 작은 테이블	215
20.1	간단한 해법: 더 큰 페이지	215
20.2	하이브리드 접근 방법: 페이징과 세그먼트	216
20.3	멀티 레벨 페이지 테이블	219
20.4	역 페이지 테이블	226
20.5	페이지 테이블을 디스크로 스와핑하기	227

20.6	요약	227
	참고 문헌	228
	숙제	229
	문제	229
제 21 장	물리 메모리 크기의 극복: 메커니즘	231
21.1	스왑 공간	232
21.2	Present Bit	233
21.3	페이지 폴트	234
21.4	메모리에 빈 공간이 없으면?	235
21.5	페이지 폴트의 처리	236
21.6	교체는 실제 언제 일어나는가	237
21.7	요약	238
	참고 문헌	239
제 22 장	물리 메모리 크기의 극복: 정책	241
22.1	캐시 관리	241
22.2	최적 교체 정책	242
22.3	간단한 정책: FIFO	244
22.4	또 다른 간단한 정책: 무작위 선택	245
22.5	과거 정보의 사용: LRU	247
22.6	워크로드에 따른 성능 비교	248
22.7	과거 이력 기반 알고리즘의 구현	252
22.8	LRU 정책 근사하기	252
22.9	갱신된 페이지(Dirty Page)의 고려	254
22.10	다른 VM 정책들	254
22.11	쓰래싱(Thrashing)	255
22.12	요약	255
	참고 문헌	256
	숙제	258
	문제	258
제 23 장	VAX/VMS 가상 메모리 시스템	259
23.1	배경	259
23.2	메모리 관리 하드웨어	260
23.3	실제 주소 공간	261
23.4	페이지 교체	262
23.5	그 외의 VM 기법들	263
23.6	요약	265
	참고 문헌	267

제 24 장 메모리 가상화를 정리하는 대화	269
제 II 편 병행성	273
제 25 장 병행성에 관한 대화	275
제 26 장 병행성: 개요	277
26.1 예제: 쓰레드 생성	278
26.2 훨씬 더 어려운 이유: 데이터의 공유	280
26.3 제어 없는 스케줄링	283
26.4 원자성에 대한 바람	285
26.5 또 다른 문제: 상대 기다리기	287
26.6 정리: 왜 운영체제에서?	287
참고 문헌	288
숙제	290
문제	290
제 27 장 막간: 쓰레드 API	293
27.1 쓰레드 생성	293
27.2 쓰레드 종료	295
27.3 락	297
27.4 컨디션 변수	299
27.5 컴파일과 실행	300
27.6 요약	301
참고 문헌	302
제 28 장 락	305
28.1 락: 기본 개념	305
28.2 Pthread 락	306
28.3 락 구현	306
28.4 락의 평가	307
28.5 인터럽트 제어	307
28.6 Test-And-Set (Atomic Exchange)	309
28.7 진짜 돌아가는 스핀 락의 구현	311
28.8 스핀 락 평가	313
28.9 Compare-And-Swap	313
28.10 Load-Linked 그리고 Store-Conditional	314
28.11 Fetch-And-Add	316
28.12 요약: 과도한 스핀	317
28.13 간단한 접근법: 무조건 양보!	318

28.14	큐의 사용: 스펜 대신 잠자기	319
28.15	다른 운영체제, 다른 지원	321
28.16	2단계 락	321
28.17	요약	322
	참고 문헌	323
	속제	325
	문제	325
제 29장 락 기반의 병행 자료 구조		327
29.1	병행 카운터	327
29.2	병행 연결 리스트	331
29.3	병행 큐	335
29.4	병행 해시 테이블	335
29.5	요약	336
	참고 문헌	339
제 30장 컨디션 변수		341
30.1	정의와 루틴들	342
30.2	생산자/소비자(유한 버퍼) 문제	345
30.3	컨디션 변수 사용 시 주의점	352
30.4	요약	354
	참고 문헌	355
제 31장 세마포어		357
31.1	세마포어: 정의	357
31.2	이진 세마포어(락)	359
31.3	컨디션 변수로서의 세마포어	361
31.4	생산자/소비자(유한 버퍼) 문제	362
31.5	Reader-Writer 락	367
31.6	식사하는 철학자	369
31.7	세마포어 구현	371
31.8	요약	372
	참고 문헌	373
제 32장 병행성 관련 오류		375
32.1	오류의 종류	375
32.2	비 교착 상태 오류	376
32.3	교착 상태 오류	379
32.4	요약	387
	참고 문헌	388

제 33 장 이벤트 기반의 병행성(고급)	391
33.1 기본 개념: 이벤트 루프	391
33.2 중요 API: <code>select ()</code> (또는 <code>poll ()</code>)	392
33.3 <code>select ()</code> 의 사용	393
33.4 왜 간단한가? 락이 필요 없음	394
33.5 문제: 블로킹 시스템 콜(Blocking System Call)	395
33.6 해법: 비동기 I/O	395
33.7 또 다른 문제점: 상태 관리	396
33.8 이벤트 사용의 어려움	398
33.9 요약	399
참고 문헌	400
제 34 장 병행성을 정리하는 대화	403
제 III 편 영속성	405
제 35 장 영속성에 관한 대화	407
제 36 장 I/O 장치	409
36.1 시스템 구조	409
36.2 표준 장치	410
36.3 표준 방식	411
36.4 인터럽트를 이용한 CPU 오버헤드 개선	412
36.5 DMA를 이용한 효율적인 데이터 이동	413
36.6 디바이스와 상호작용하는 방법	414
36.7 운영체제에 연결하기: 디바이스 드라이버	415
36.8 사례 연구: 간단한 IDE 디스크 드라이버	417
36.9 역사상의 기록	418
36.10 요약	420
참고 문헌	421
제 37 장 하드 디스크 드라이브	423
37.1 인터페이스	423
37.2 기본 구조	424
37.3 간단한 디스크 드라이브	424
37.4 I/O 시간 계산	429
37.5 디스크 스케줄링	431
37.6 요약	436
참고 문헌	437

제 38장 Redundant Array of Inexpensive Disk (RAID)	441
38.1 인터페이스와 RAID의 내부	442
38.2 결합 모델	443
38.3 RAID의 평가 방법	443
38.4 RAID 레벨 0: 스트라이핑	444
38.5 RAID 레벨 1: 미러링	448
38.6 RAID 레벨 4: 패리티를 이용한 공간 절약	451
38.7 RAID 레벨 5: 순환 패리티	455
38.8 RAID 비교: 정리	456
38.9 RAID와 관련된 다른 흥미로운 주제들	457
38.10 요약	457
참고 문헌	458
숙제	460
문제	460
제 39장 막간: 파일과 디렉터리	461
39.1 파일과 디렉터리	461
39.2 파일 시스템 인터페이스	463
39.3 파일의 생성	463
39.4 파일의 읽기와 쓰기	464
39.5 비 순차적 읽기와 쓰기	466
39.6 <code>fsync()</code> 를 이용한 즉시 기록	467
39.7 파일 이름 변경	468
39.8 파일 정보 추출	469
39.9 파일 삭제	469
39.10 디렉터리 생성	470
39.11 디렉터리 읽기	471
39.12 디렉터리 삭제하기	472
39.13 하드 링크	472
39.14 심볼릭 링크	474
39.15 파일 시스템 생성과 마운트	475
39.16 요약	477
참고 문헌	478
숙제	479
문제	479
제 40장 파일 시스템 구현	481
40.1 생각하는 방법	481
40.2 전체 구성	482

40.3	파일 구성: 아이노드	485
40.4	디렉터리 구조	489
40.5	빈 공간의 관리	491
40.6	실행 흐름: 읽기와 쓰기	491
40.7	캐싱과 버퍼링	495
40.8	요약	497
	참고 문헌	499
	속제	501
	문제	501
제 41 장 지역성과 Fast File System		503
41.1	문제: 낮은 성능	503
41.2	FFS: 디스크에 대한 이해가 해답이다	505
41.3	파일 시스템 구조: 실린더 그룹	505
41.4	파일과 디렉터리 할당 정책	506
41.5	파일 접근의 지역성 측정	507
41.6	대용량 파일 예외 상황	508
41.7	FFS에 대한 기타 사항	510
41.8	요약	512
	참고 문헌	513
제 42 장 크래시 일관성: FSCK와 저널링		515
42.1	예제	516
42.2	해법 1: 파일 시스템 검사기	519
42.3	해법 2: 저널링(또는 Write-Ahead Logging)	521
42.4	해법 3: 그 외 방법	532
42.5	요약	533
	참고 문헌	534
제 43 장 로그 기반 파일 시스템		537
43.1	디스크에 순차적으로 쓰기	538
43.2	순차적이면서 효율적으로 쓰기	539
43.3	적절한 버퍼의 크기는?	540
43.4	문제: 아이노드 찾기	541
43.5	간접 계층을 이용한 해법: 아이노드 맵	541
43.6	최종 완성: 체크포인트 영역	542
43.7	디스크에서 읽기: 요약	543
43.8	디렉터리 관리 방법은?	543
43.9	새로운 문제: 가비지 컬렉션	544
43.10	블럭의 최신 여부 판단	546

43.11	정책: 어떤 블럭을 언제 정리하는가	547
43.12	크래시로부터의 복구와 로그	548
43.13	요약	549
	참고 문헌	550
제 44 장	데이터 무결성과 보호	553
44.1	디스크 오류 모델	553
44.2	숨어있는 섹터 에러 (Latent Sector Error)	555
44.3	손상 검출: 체크섬	556
44.4	체크섬의 활용	559
44.5	새로운 문제: 잘못된 위치에 기록	560
44.6	마지막 문제: 기록 작업의 손실	561
44.7	Scrubbing	562
44.8	체크섬 오버헤드	562
44.9	요약	563
	참고 문헌	564
제 45 장	영속성을 정리하는 대화	567
제 46 장	분산에 관한 대화	569
제 47 장	분산 시스템	571
47.1	통신의 기본	572
47.2	신뢰할 수 없는 통신 계층	573
47.3	신뢰할 수 있는 통신 계층	574
47.4	통신 추상화	577
47.5	Remote Procedure Call (RPC)	578
47.6	요약	583
	참고 문헌	585
제 48 장	Sun 사의 네트워크 파일 시스템 (NFS)	587
48.1	기본적인 분산 파일 시스템	588
48.2	NFS에 대하여	589
48.3	핵심: 단순하고 빠른 서버 크래시 복구	589
48.4	빠른 크래시 복구의 열쇠: 상태를 유지하지 않음	590
48.5	NFSv2 프로토콜	591
48.6	프로토콜에서 분산 파일 시스템으로	593
48.7	서버의 고장을 멍등연산으로 처리하기	595
48.8	성능 개선하기: 클라이언트 측 캐싱	597
48.9	캐시 일관성 문제	598

48.10 NFS의 캐시 일관성 기법에 대한 평가	599
48.11 서버 측 쓰기 버퍼링의 의미	600
48.12 요약	601
참고 문헌	602
제 49 장 Andrew 파일 시스템 (AFS)	605
49.1 AFS 버전 1	605
49.2 버전 1의 문제점	606
49.3 프로토콜의 개선	608
49.4 AFS 버전 2	608
49.5 캐시 일관성	610
49.6 크래시 복구	611
49.7 AFSv2의 확장성과 성능	613
49.8 AFS: 그 외의 개선점들	615
49.9 요약	616
참고 문헌	617
속제	618
문제	618
제 50 장 분산을 정리하는 대화	619
색인	621
여담	641
팁	643
핵심 질문	645