

2022학년도 2학기

운영체제

# -REPORT-



수업명	운영체제
과제 이름	Assignment5
담당 교수님	최상호 교수님
학번	2018204058
이름	김민교

## 1. Introduction :

디스크 I/O scheduler를 바꿔가면서 파일 입출력 요청을 진행한다. I/O scheduler를 바꾸는 명령어를 익히고 실제로 리눅스 명령어를 통해 바꾼다. CFQ, NOOP, Deadline 스케줄러를 사용하고 세 개의 차이를 알아본다. 스케줄러 별 성능을 측정해보고 결과를 확인한다. 그래프와 표를 생성해보고 스케줄러의 차이를 눈으로 본다. 스케줄링 별 각 장단점을 알 수 있다.

## 2. Conclusion & Analysis :

### I/O Scheduler

디스크 스케줄링의 목표 : 탐색 시간과 회전 지연시간의 최소화를 위함이다.

#### [Deadline I/O scheduler]

: 4개의 큐를 사용한다. 데드라인이 지난 요청이 없을 경우에 정렬된 큐에서 요청을 꺼내서 처리한다.

#### [CFQ(Complete Fair Queueing) I/O scheduler]

: 입출력을 요청하는 모든 프로세스들에 대해 디스크 I/O 대역폭을 공평하게 할당하는 것을 보장하는 기법이다.

#### [NOOP(NO Operation) I/O scheduler]

: 인접한 요청 병합만 수행하고 그 외에 아무 작업을 하지 않음. Random access하는 device를 위한 스케줄러이다.

#### IOZone 설치 -

#### [테스트 연산]

1. write/re-write
2. read/re-read
3. random-read/write
4. read-backwards

read-backwards를 선택한 이유 :

스케줄러 확인

```
os2018204058@ubuntu:~/assign5$ cat /sys/block/sda/queue/scheduler  
noop deadline [cfq]
```

noop으로 변경

```
os2018204058@ubuntu:~/assign5$ echo noop | sudo tee /sys/block/sda/queue/scheduler
[sudo] password for os2018204058:
noop
os2018204058@ubuntu:~/assign5$ cat /sys/block/sda/queue/scheduler
[noop] deadline cfq
```

실험 전 캐시 및 버퍼를 지우고 실험에 영향을 주는 요소를 제거

```
os2018204058@ubuntu:~/iozone_test$ rm -rf ~/iozone_test
os2018204058@ubuntu:~/iozone_test$ sync
os2018204058@ubuntu:~/iozone_test$ echo 3 | sudo tee /proc/sys/vm/drop_caches
3
```

file size : 1m

buffer cache : 거치지 않고 연산을 수행합니다.

record size : 8k, 16k, 32k, 64k, 128k, 256k, 512k, 8m, 16m

thread or process : 1개

thread or process의 파일 경로 : ~/iozone\_test

로 설정하고 iozone 프로그램을 실행했습니다.

이름	수정된 날짜	유형	크기
cfq_8k.xls	2022-12-08 오후 1:52	Microsoft Excel 97...	4KB
cfq_8k_1.xls	2022-12-08 오후 1:52	Microsoft Excel 97...	4KB
cfq_8k_2.xls	2022-12-08 오후 1:52	Microsoft Excel 97...	4KB
cfq_8k_3.xls	2022-12-08 오후 1:52	Microsoft Excel 97...	4KB
cfq_8k_4.xls	2022-12-08 오후 1:52	Microsoft Excel 97...	4KB
cfq_8m.xls	2022-12-08 오후 1:52	Microsoft Excel 97...	4KB
cfq_8m_1.xls	2022-12-08 오후 1:52	Microsoft Excel 97...	4KB
cfq_8m_2.xls	2022-12-08 오후 1:52	Microsoft Excel 97...	4KB
cfq_8m_3.xls	2022-12-08 오후 1:52	Microsoft Excel 97...	4KB
cfq_8m_4.xls	2022-12-08 오후 1:52	Microsoft Excel 97...	4KB

리눅스 환경에서 평균을 내고 표를 만들기가 어려워 윈도우 환경으로 xls파일을 옮겨서 진행했습니다.

iozone -R -i 0 -i 1 -i 2 -i 3 -l -r -s 1m -t 1 -F /home/os2018204058/iozone_test -b cfq_8k_1.xls 8k									
Throughput report Y-axis is type of test X-axis is number of processes									
Record size = 4 kBytes									
Output is in kBytes/sec									
Initial write	1950657								
Rewrite	2586075								
Read	5120336								
Re-read	4926519								
Reverse Read	4589593								
Random read	4904019								
Random write	2504638								

.xls 파일 내용입니다.

	A	B	C	D	E	F	G	H	I
1			Initial write	Rewrite	Read	Re-read	Reverse Read	indom rea	Random write
2	CFQ	8k	1854632	1845071	3709575	4396952	4079544.25	4876180	2598591.75
3			1950656.875	2586075	5120336	4926519	4589593	4904019	2504638
4			1942715.875	2504638	4854136	4920875	4142499.75	4249053	2461573.5
5			1868347.25	2260740	5018624	4339203	4594502.5	4926519	2061117
6			1724328.5	1950657	4415031	4526709	4146499.25	4854136	1999700.125
7		평균	1868136.1	2229436	4623540	4622051	4310527.75	4761981	2325124.075
8		8m	1796451.875	2666347	4995277	4304413	4574926.5	4550690	2198257.25
9			1720874.125	2723841	4570059	4356809	4609295	4832291	2579861
10			1774922.375	2744729	4374559	4805258	3866533.5	4574927	2572136
11			1822368	2659743	4339203	4920875	1544504.625	4720752	2592318
12			1882266.875	2646631	4832291	4876180	4433259.5	4926519	2503178.25
13		평균	1799376.65	2688258	4622277	4652707	3805703.825	4721036	2489150.1
14		16k	135160.4375	2659743	4550690	4898425	4282950.5	4488861	2472911.75
15			1861867.875	2541693	4512441	4876180	3970183.75	4356809	2426801.75
16			1393200.75	2611231	4674511	4674511	4029784.75	5096035	1943595
17			1729188.125	2791105	5303701	4415031	3778101.5	4920875	2559871.75
18			1848247.25	2540190	4966396	4898425	4570058.5	4249053	2612819.5
19		평균	1393532.888	2628792	4801548	4752514	4126215.8	4622326	2403199.95
20		16m	1893052.125	2744729	5226256	4614247	2813041.25	4232305	2586074.5
21			1899750.75	1541733	4783850	4810641	4199201.5	4920875	1980337.5
22			1861867.875	2572136	4995277	4995277	4470172.5	4810641	2409105.75
23			1665487	2619193	4995277	5065981	4507705	4589593	1924436
24			1917562.5	2653171	5071963	4826860	4594502.5	4550690	2480051.5
25		평균	1847544.05	2426192	5014524	4862601	4116924.55	4620821	2276001.05
26		32k	1701108.125	2316838	4805258	4943530	4199201.5	4614247	2565989.25
27			1906497	2672985	5065981	4674511	4415031	4783850	2553783.25
28			1738286.125	2701568	4550690	4848656	3894582.25	5042191	2331933
29			1872419.875	1882267	4570059	4589593	5169641	5336651	2760606.5
30		평균	1444266.25	2708382	5144871	4783850	4720752	5089996	2534194.5
31		64k	1786736.5	2559872	4920875	4832291	4130548	3582705	2553783.25
32			1679160.875	2472912	3794792	5096035	4609295	4654249	2415881.25
33			1917562.5	1896396	4474830	4232305	4589593	4736370	1868347.25
34			1771262.5	2993414	5529011	5572049	3579719.25	4715569	2837198.5
35		평균	1910737.875	2694787	5303701	4629167	4178773.75	4920875	2169388.25
36		128k	1751042.625	2686360	4943530	4282951	4789183.5	4762631	2328140.75
37			1783768.25	2353657	4995277	4415031	4282950.5	4949227	2353657.25
38			1928757.4375	3333734	5943684	4558688	4674544	4976688	3558874.75

엑셀로 record size, 스케줄러, 연산 등을 나누어 평균값을 구했습니다.

[CFQ 스케줄러 성능]

		Initial write	Rewrite	Read	Re-read	Reverse Read	Random read	Random write
CFQ	8k	1868136.1	2229436	4623540	4622051	4310527.75	4761981.3	2325124.075
	16k	1393532.89	2628792	4801548	4752514	4126215.8	4622326.4	2403199.95
	32k	1444266.25	2708382	5144871	4783850	4720752	5089995.5	2534194.5
	64k	1910737.88	2694787	5303701	4629167	4178773.75	4920874.5	2169388.25
	128k	1876096.9	2566995	5040704	4708917	4509090.55	4959973.8	2410195.05
	256k	1835314.88	2689500	4864728	4875065	4113285.35	4900809.6	2326458.325
	512k	1881378.4	2495182	5094543	4646999	4326237.05	4835605	2513758.05
	8m	1799376.65	2688258	4622277	4652707	3805703.825	4721035.6	2489150.1
	16m	1847544.05	2426192	5014524	4862601	4116924.55	4620820.6	2276001.05
	평균값	1761820.44	2569725	4945604	4725986	4245278.958	4825935.811	2383052.15

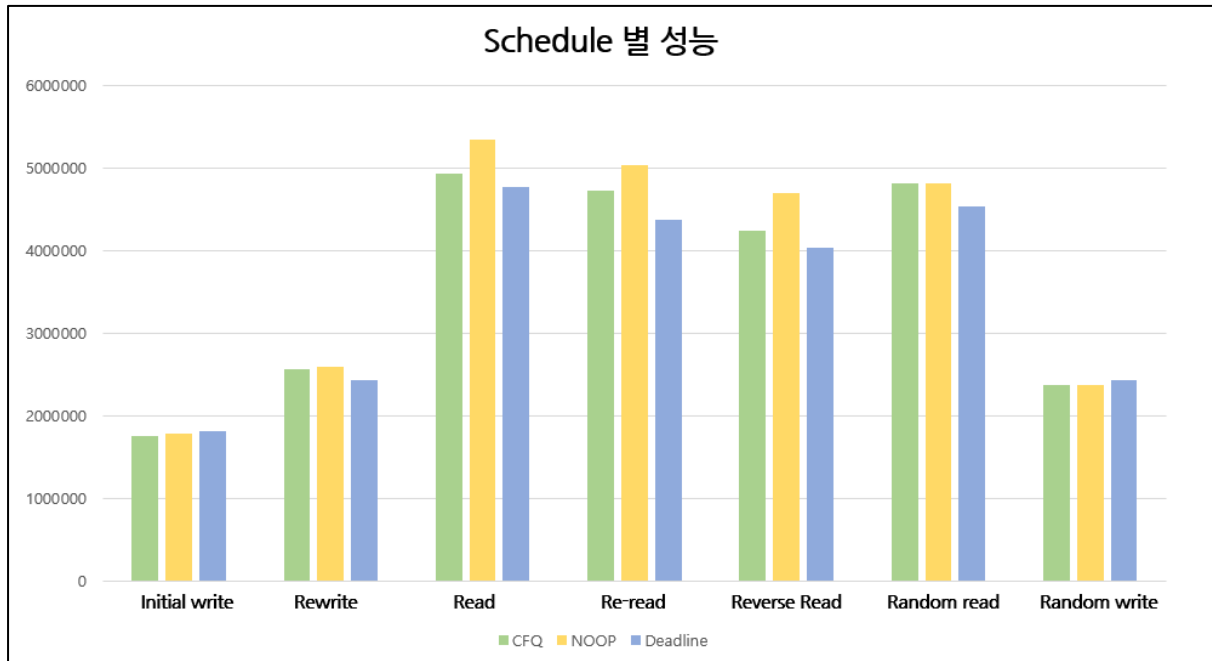
[NOOP 스케줄러 성능]

		Initial write	Rewrite	Read	Re-read	Reverse Read	Random read	Random write
NOOP	8k	1603662.35	3457090	8567645	7277032	6785605.5	5562585	1806571.917
	16k	1787123.08	2518008	4997469	4816038	4539989.833	4671197.667	2601429.5
	32k	1795144.29	2372279	4807260	4884269	4388265.167	4310488.167	2396083.583
	64k	1728033.25	2565047	5330290	5074418	4373071.5	5057162	2523141.75
	128k	1722827.96	2331396	4922969	4613444	4628122.167	4774519.667	2305305.083
	256k	1820911.08	2440622	5008485	4922842	4423054.833	4690517.5	2423280
	512k	1835879.75	2450027	4765574	4541044	4479769.5	4772827	2472863.417
	8m	1877386.5	2579639	4935580	4580364	4315154.25	4723677.667	2508047.917
	16m	1862108.29	2594769	4763324	4669049	4353286.333	4819856	2410717.75
	평균값	1781452.95	2589875	5344288	5042056	4698479.898	4820314.519	2383048.991

[Deadline 스케줄러 성능]

		Initial write	Rewrite	Read	Re-read	Reverse Read	Random read	Random write
Deadline	8k	1793668.5	2337765	4963013	4025231	3423926.292	4534794.167	2472048.333
	16k	1894334.42	2353758	4665992	3857804	4580311.667	4207613.583	2319060.417
	32k	1828440.75	2519864	4817313	4385468	4385517.333	4464289.25	2361234.792
	64k	1841019.5	2380511	4926278	4983625	4401127.333	4803767.167	2535323.417
	128k	1764166.63	2652638	4716650	4854236	3703629.5	4621652.167	2464153.417
	256k	1921369.67	2704676	4841372	3873021	3749981.417	4730478.167	2419385.667
	512k	1760252.63	2337273	4887638	4737641	3967330.917	4286987.583	2381841.875
	8m	1875902.08	2171279	4503643	4276835	4316603.333	4561600.333	2463044.083
	16m	1726364.63	2504561	4620143	4439308	3830545.333	4660603	2488144.708
	평균값	1822835.42	2440258	4771338	4381463	4039885.903	4541309.491	2433804.079

[스케줄 별 성능 그래프]



initial write는 CFQ가 가장 빨랐다. Deadline은 읽기 우선 정책이라서 그런지 쓰기가 조금 오래 걸리는 듯 하다. NOOP

Rewrite는 Deadline이 가장 빨랐다. 초기 write에 비해 Rewrite가 CFQ일 때 크게 늘어났다.

Read를 할 때에는 NOOP이 가장 느렸다.

Re-read 할 때에도 NOOP이 가장 느렸다.

Reverse - Read를 할 때에도 NOOP이 가장느렸다.

Random Read

## [Reference]

<https://rhlinux.tistory.com/26>

<https://duksoo.tistory.com/entry/Linux-IO-%EC%8A%A4%EC%BC%80%EC%A4%84%EB%9F%AC%EC%9D%98-%EC%A2%85%EB%A5%98-%EC%99%80-%EC%9A%B4%EC%98%81%ED%99%98%EA%B2%BD%EC%97%90-%EB%94%B0%EB%A5%B8-%EC%B5%9C%EC%A0%81%ED%99%94-%EB%B0%A9%EB%B2%95>

<https://develoid.github.io/android/Android-IO-Schedulers.html>