

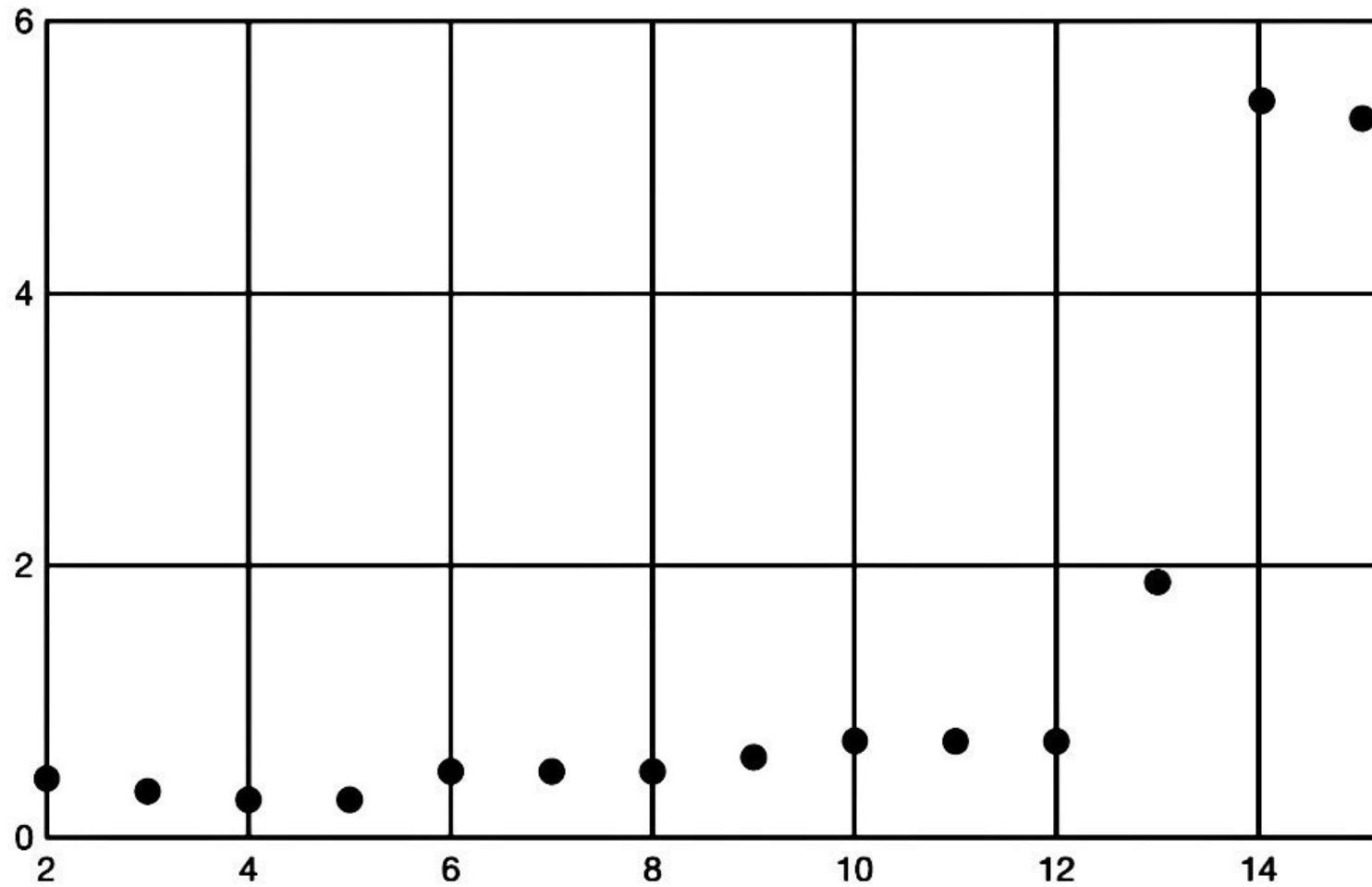
20182/3354 5월 1

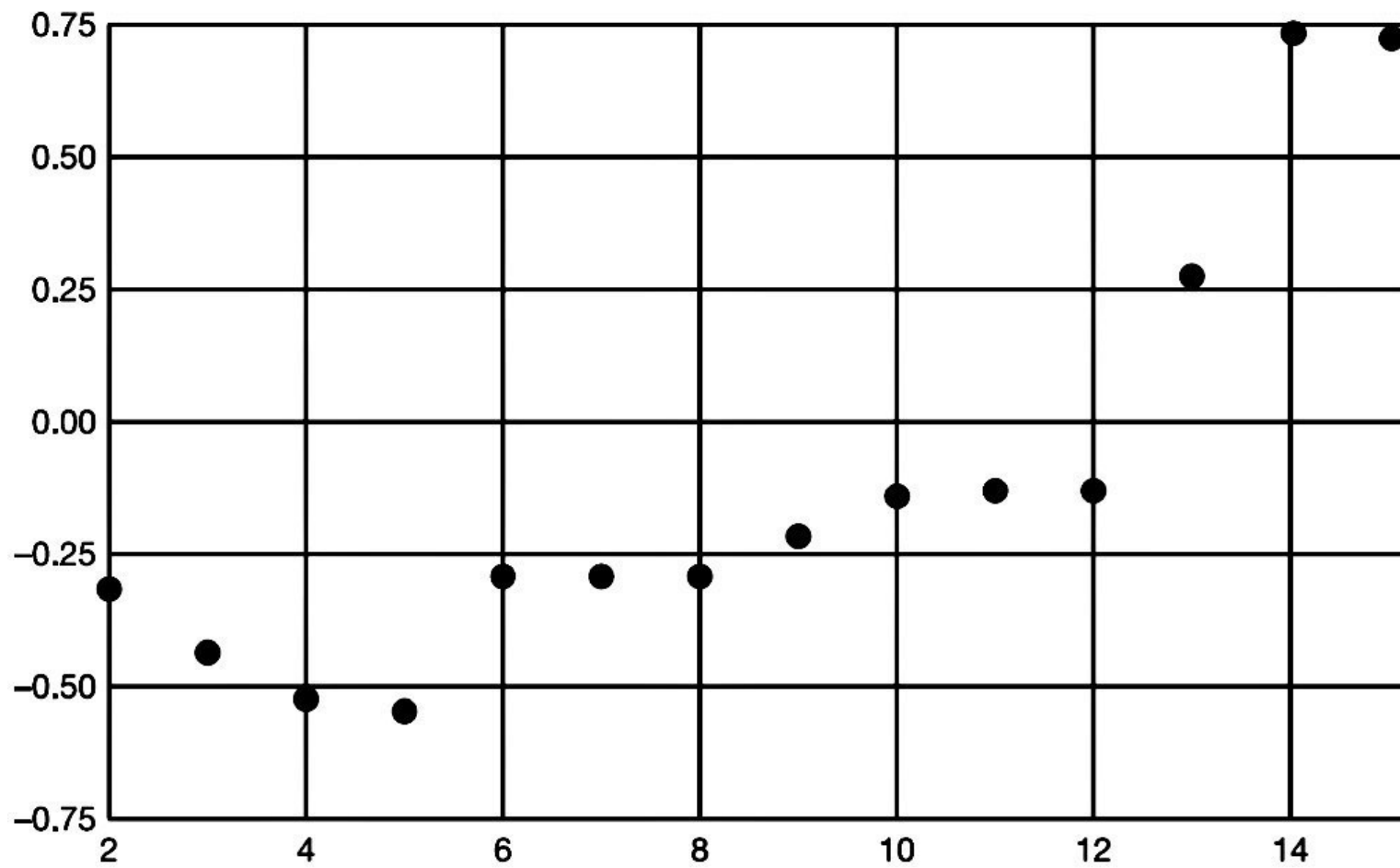
< Lab5 >

1.

그래프를 보며 대략 각 캐시의 사이클 상계를 해서 계산을 변경하고 있음  
왔수있지.

```
$ for i in 4 8 16 32 64 128 256 512 1024 2048 4096 8192 16384 32768 ; do ./cache  
$ i ; done  
0.476930  
0.363404  
0.302903  
0.279884  
0.504577  
0.502791  
0.503517  
0.602227  
0.726228  
0.730371  
0.728870  
1.898528  
5.412608  
5.282390  
$
```





210.

\$ free

	total	used	free	shared	buff/cache	available
Mem:	32941348	203820	32441740	9664	295788	32272416
Swap:	0	0	0			

\$ time cat testfile >/dev/null

real 0m2.002s

user 0m0.000s

sys 0m0.468s

\$ free

	total	used	free	shared	buff/cache	available
Mem:	32941348	205336	31385744	9664	1350268	32244916
Swap:	0	0	0			

\$

2차 장치가 끝났는데 처음 읽는거 저장장치에 걸리고 값을 할 수 있다.  
경다 시간 2초에 프로세스 혹은 프로세스의 처리를 요청하는 커널이 CPU를 사용한 시간  
0.468초. 전체의 3/4 정도인 1.554초는 저장장치로부터 읽기가 끝나기를 기다리는데  
걸린 시간이다. 또한 특정 컴퓨터 시스템 전체 페이지 캐시가 1Gi 정도 증가했을  
알 수 있다. 이제 `tedsize`의 페이지 캐시이다.

```
$ time cat testfile >/dev/null
```

```
real 0m0.100s
```

```
user 0m0.000s
```

```
sys 0m0.104s
```



**\$ free**

	total	used	free	shared	buff/cache	available
Mem:	32941348	205036	31385760	9664	1350552	32245312
Swap:						

첫번째로 RAM값도 크게 줄었다. 두번째 저장장치 하에 전하가 없  
퍼/버지 캐시에서 데이터를 복사했기 때문으로. 시스템이 사용중인 페이지  
캐시의 용량도 변하지 X.

```
$ ./read-twice.sh
Thu Dec 28 13:04:04 JST 2017: start file creation
1024+0 records in
1024+0 records out
1073741824 bytes (1.1 GB, 1.0 GiB) copied, 2.98329 s, 360 MB/s
Thu Dec 28 13:04:07 JST 2017: end file creation
Thu Dec 28 13:04:07 JST 2017: sleep 3 seconds
Thu Dec 28 13:04:10 JST 2017: start 1st read
Thu Dec 28 13:04:12 JST 2017: end 1st read
Thu Dec 28 13:04:12 JST 2017: sleep 3 seconds
Thu Dec 28 13:04:15 JST 2017: start 2st read
Thu Dec 28 13:04:16 JST 2017: end 2nd read
```

13:04:03	0.00	0.00	0.00	0.00	2.00	0.00	0.00
0.00	0.00						
13:04:04	0.00	0.00	0.00	0.00	2.00	0.00	0.00
0.00	0.00						
13:04:05	256.00	206848.00	749.00	0.00	240.00	0.00	0.00
0.00	0.00						
13:04:06	1552.00	372736.00	0.00	0.00	3.00	0.00	0.00
0.00	0.00						
13:04:07	1216.00	331776.00	0.00	0.00	1.00	0.00	0.00
0.00	0.00						
13:04:08	416.00	137216.00	363.00	0.00	506.00	0.00	0.00
0.00	0.00						
13:04:09	0.00	0.00	0.00	0.00	3.00	0.00	0.00
0.00	0.00						
13:04:10	0.00	0.00	0.00	0.00	2.00	0.00	0.00
0.00	0.00						
13:04:11	286208.00	0.00	275.00	0.00	212.00	0.00	0.00
0.00	0.00						
13:04:12	524288.00	0.00	0.00	0.00	106.00	0.00	0.00
0.00	0.00						
13:04:13	238080.00	0.00	361.00	0.00	288.00	0.00	0.00
0.00	0.00						
13:04:14	3312.00	24252.00	0.00	0.00	890.00	0.00	0.00
0.00	0.00						
13:04:15	0.00	0.00	0.00	0.00	1.00	0.00	0.00
0.00	0.00						
13:04:16	112.00	0.00	538.00	0.00	263182.00	0.00	0.00
0.00	0.00						
13:04:17	0.00	0.00	0.00	0.00	1.00	0.00	0.00
0.00	0.00						
13:04:18	0.00	0.00	0.00	0.00	2.00	0.00	0.00
0.00	0.00						

결과를 보면 파일 작성할 때 16진수의 페이지 어드레스가 발생하고 있음을 알 수 있는데  
첫번째 파일을 읽을 때는 총 16개 페이지가 발생하고 있다.  
여기서 저장장치에서 페이지 캐싱이 데이터를 읽어들이고 있음을 알 수 있다.

(두 파일을 읽을 때 페이지가 발생하지 않음) → 시스템의 다른 부분이 원인.

```
$ mount | grep "on / "  
/dev/sda5 on / type btrfs (rw,relatime,ssd,space_cache,subvolid=257,subvol=/@)
```

내환경에서는 /dev/sda5가 존재하는 것을 보았다.