

UNIX Team Project

19011583 최건우
20003212 정병준
20011914 정기남
22011631 박지우

2024 – 2 UNIX Programming

Contents

1. 128 x 128 Integer Array Partitioning

- 4by4
- 8by8

2. Client → Client Communication

- 8by8
- 4by4

3. Client → Server Communication

128 x 128 Integer Array Partitioning

4 by 4 / 8 by 8

4by4 Partitioning

128 x 128 정수 배열



4by4 Partitioning

```
for (sm = 0; sm < 8; sm++) {
    for (block_offset = 0; block_offset < 2; block_offset++) {

        // 각 SM에 맞는 블록을 쓰기
        int block_num;
        block_num = 8 * block_offset + sm;

        // 각 블록의 시작 행과 열
        int start_row = (block_num / 4) * BLOCK_SIZE;
        int start_col = (block_num % 4) * BLOCK_SIZE;

        int block[BLOCK_SIZE][BLOCK_SIZE];

        for (i = 0; i < BLOCK_SIZE; i++) {
            for (j = 0; j < BLOCK_SIZE; j++) {
                block[i][j] = matrix[start_row + i][start_col + j];
            }
        }

        // SM 파일에 쓰기
        if (write(sm_fds[sm], block, sizeof(block)) != sizeof(block)) {
            perror("SM 파일 쓰기 실패");
            return 1;
        }
    }
}
```

SM0 - 4by4 Partitioning

Front

```
[s22011631@sce 4by4]$ od -i SM1
0000000      0      1      2      3
0000020      4      5      6      7
0000040      8      9     10     11
0000060     12     13     14     15
0000100     16     17     18     19
0000120     20     21     22     23
0000140     24     25     26     27
0000160     28     29     30     31
0000200    128    129    130    131
0000220    132    133    134    135
0000240    136    137    138    139
0000260    140    141    142    143
0000300    144    145    146    147
0000320    148    149    150    151
0000340    152    153    154    155
0000360    156    157    158    159
0000400    256    257    258    259
0000420    260    261    262    263
0000440    264    265    266    267
0000460    268    269    270    271
0000500    272    273    274    275
```

Back, SIZE = 8192

```
0017300    11920    11921    11922    11923
0017320    11924    11925    11926    11927
0017340    11928    11929    11930    11931
0017360    11932    11933    11934    11935
0017400    12032    12033    12034    12035
0017420    12036    12037    12038    12039
0017440    12040    12041    12042    12043
0017460    12044    12045    12046    12047
0017500    12048    12049    12050    12051
0017520    12052    12053    12054    12055
0017540    12056    12057    12058    12059
0017560    12060    12061    12062    12063
0017600    12160    12161    12162    12163
0017620    12164    12165    12166    12167
0017640    12168    12169    12170    12171
0017660    12172    12173    12174    12175
0017700    12176    12177    12178    12179
0017720    12180    12181    12182    12183
0017740    12184    12185    12186    12187
0017760    12188    12189    12190    12191
0020000
[s22011631@sce 4by4]$ du -b SM1
8192    SM1
```

SM1 - 4by4 Partitioning

Front

```
[s22011631@sce 4by4]$ od -i SM2
0000000      32      33      34      35
00000020     36      37      38      39
00000040     40      41      42      43
00000060     44      45      46      47
00000100     48      49      50      51
00000120     52      53      54      55
00000140     56      57      58      59
00000160     60      61      62      63
00000200    160     161     162     163
00000220    164     165     166     167
00000240    168     169     170     171
00000260    172     173     174     175
00000300    176     177     178     179
00000320    180     181     182     183
00000340    184     185     186     187
00000360    188     189     190     191
00000400    288     289     290     291
00000420    292     293     294     295
00000440    296     297     298     299
00000460    300     301     302     303
00000500    304     305     306     307
00000520    308     309     310     311
```

Back, SIZE = 8192

```
0017300      11952     11953     11954     11955
0017320      11956     11957     11958     11959
0017340      11960     11961     11962     11963
0017360      11964     11965     11966     11967
0017400      12064     12065     12066     12067
0017420      12068     12069     12070     12071
0017440      12072     12073     12074     12075
0017460      12076     12077     12078     12079
0017500      12080     12081     12082     12083
0017520      12084     12085     12086     12087
0017540      12088     12089     12090     12091
0017560      12092     12093     12094     12095
0017600      12192     12193     12194     12195
0017620      12196     12197     12198     12199
0017640      12200     12201     12202     12203
0017660      12204     12205     12206     12207
0017700      12208     12209     12210     12211
0017720      12212     12213     12214     12215
0017740      12216     12217     12218     12219
0017760      12220     12221     12222     12223
0020000
[s22011631@sce 4by4]$ du -b SM2
8192      SM2
```

SM2 - 4by4 Partitioning

Front

```
[s22011631@sce 4by4]$ clear
[s22011631@sce 4by4]$ od -i SM3
0000000      64      65      66      67
0000020      68      69      70      71
0000040      72      73      74      75
0000060      76      77      78      79
0000100      80      81      82      83
0000120      84      85      86      87
0000140      88      89      90      91
0000160      92      93      94      95
0000200     192     193     194     195
0000220     196     197     198     199
0000240     200     201     202     203
0000260     204     205     206     207
0000300     208     209     210     211
0000320     212     213     214     215
0000340     216     217     218     219
0000360     220     221     222     223
0000400     320     321     322     323
0000420     324     325     326     327
```

Back, SIZE = 8192

```
0017300      11984    11985    11986    11987
0017320      11988    11989    11990    11991
0017340      11992    11993    11994    11995
0017360      11996    11997    11998    11999
0017400      12096    12097    12098    12099
0017420      12100    12101    12102    12103
0017440      12104    12105    12106    12107
0017460      12108    12109    12110    12111
0017500      12112    12113    12114    12115
0017520      12116    12117    12118    12119
0017540      12120    12121    12122    12123
0017560      12124    12125    12126    12127
0017600      12224    12225    12226    12227
0017620      12228    12229    12230    12231
0017640      12232    12233    12234    12235
0017660      12236    12237    12238    12239
0017700      12240    12241    12242    12243
0017720      12244    12245    12246    12247
0017740      12248    12249    12250    12251
0017760      12252    12253    12254    12255
0020000
[s22011631@sce 4by4]$ du -b SM3
8192    SM3
```

SM3 - 4by4 Partitioning

Front

```
[s22011631@sce 4by4]$ od -i SM4
0000000      96      97      98      99
00000020     100     101     102     103
00000040     104     105     106     107
00000060     108     109     110     111
00000100     112     113     114     115
00000120     116     117     118     119
00000140     120     121     122     123
00000160     124     125     126     127
00000200     224     225     226     227
00000220     228     229     230     231
00000240     232     233     234     235
00000260     236     237     238     239
00000300     240     241     242     243
00000320     244     245     246     247
00000340     248     249     250     251
00000360     252     253     254     255
00000400     352     353     354     355
00000420     356     357     358     359
00000440     360     361     362     363
00000460     364     365     366     367
```

Back, SIZE = 8192

0017300	12016	12017	12018	12019
0017320	12020	12021	12022	12023
0017340	12024	12025	12026	12027
0017360	12028	12029	12030	12031
0017400	12128	12129	12130	12131
0017420	12132	12133	12134	12135
0017440	12136	12137	12138	12139
0017460	12140	12141	12142	12143
0017500	12144	12145	12146	12147
0017520	12148	12149	12150	12151
0017540	12152	12153	12154	12155
0017560	12156	12157	12158	12159
0017600	12256	12257	12258	12259
0017620	12260	12261	12262	12263
0017640	12264	12265	12266	12267
0017660	12268	12269	12270	12271
0017700	12272	12273	12274	12275
0017720	12276	12277	12278	12279
0017740	12280	12281	12282	12283
0017760	12284	12285	12286	12287
0020000				

```
[s22011631@sce 4by4]$ du -b SM4
8192    SM4
```

SM4 - 4by4 Partitioning

Front

```
[s22011631@sce 4by4]$ od -i SM5
0000000 4096 4097 4098 4099
0000020 4100 4101 4102 4103
0000040 4104 4105 4106 4107
0000060 4108 4109 4110 4111
0000100 4112 4113 4114 4115
0000120 4116 4117 4118 4119
0000140 4120 4121 4122 4123
0000160 4124 4125 4126 4127
0000200 4224 4225 4226 4227
0000220 4228 4229 4230 4231
0000240 4232 4233 4234 4235
0000260 4236 4237 4238 4239
0000300 4240 4241 4242 4243
0000320 4244 4245 4246 4247
0000340 4248 4249 4250 4251
0000360 4252 4253 4254 4255
0000400 4352 4353 4354 4355
0000420 4356 4357 4358 4359
0000440 4360 4361 4362 4363
0000460 4364 4365 4366 4367
```

Back, SIZE = 8192

0017300	16016	16017	16018	16019
0017320	16020	16021	16022	16023
0017340	16024	16025	16026	16027
0017360	16028	16029	16030	16031
0017400	16128	16129	16130	16131
0017420	16132	16133	16134	16135
0017440	16136	16137	16138	16139
0017460	16140	16141	16142	16143
0017500	16144	16145	16146	16147
0017520	16148	16149	16150	16151
0017540	16152	16153	16154	16155
0017560	16156	16157	16158	16159
0017600	16256	16257	16258	16259
0017620	16260	16261	16262	16263
0017640	16264	16265	16266	16267
0017660	16268	16269	16270	16271
0017700	16272	16273	16274	16275
0017720	16276	16277	16278	16279
0017740	16280	16281	16282	16283
0017760	16284	16285	16286	16287
0020000				

```
[s22011631@sce 4by4]$ du -b SM5
8192    SM5
```

SM5 - 4by4 Partitioning

Front

```
[s22011631@sce 4by4]$ od -i SM6
0000000 4128 4129 4130 4131
0000020 4132 4133 4134 4135
0000040 4136 4137 4138 4139
0000060 4140 4141 4142 4143
0000100 4144 4145 4146 4147
0000120 4148 4149 4150 4151
0000140 4152 4153 4154 4155
0000160 4156 4157 4158 4159
0000200 4256 4257 4258 4259
0000220 4260 4261 4262 4263
0000240 4264 4265 4266 4267
0000260 4268 4269 4270 4271
0000300 4272 4273 4274 4275
0000320 4276 4277 4278 4279
0000340 4280 4281 4282 4283
0000360 4284 4285 4286 4287
0000400 4384 4385 4386 4387
0000420 4388 4389 4390 4391
0000440 4392 4393 4394 4395
0000460 4396 4397 4398 4399
```

Back, SIZE = 8192

```
0017300 16048 16049 16050 16051
0017320 16052 16053 16054 16055
0017340 16056 16057 16058 16059
0017360 16060 16061 16062 16063
0017400 16160 16161 16162 16163
0017420 16164 16165 16166 16167
0017440 16168 16169 16170 16171
0017460 16172 16173 16174 16175
0017500 16176 16177 16178 16179
0017520 16180 16181 16182 16183
0017540 16184 16185 16186 16187
0017560 16188 16189 16190 16191
0017600 16288 16289 16290 16291
0017620 16292 16293 16294 16295
0017640 16296 16297 16298 16299
0017660 16300 16301 16302 16303
0017700 16304 16305 16306 16307
0017720 16308 16309 16310 16311
0017740 16312 16313 16314 16315
0017760 16316 16317 16318 16319
0020000
```

```
[s22011631@sce 4by4]$ du -b SM6
8192 SM6
```

SM6 - 4by4 Partitioning

Front

```
[s22011631@sce 4by4]$ od -i SM7
0000000 4160 4161 4162 4163
0000020 4164 4165 4166 4167
0000040 4168 4169 4170 4171
0000060 4172 4173 4174 4175
0000100 4176 4177 4178 4179
0000120 4180 4181 4182 4183
0000140 4184 4185 4186 4187
0000160 4188 4189 4190 4191
0000200 4288 4289 4290 4291
0000220 4292 4293 4294 4295
0000240 4296 4297 4298 4299
0000260 4300 4301 4302 4303
0000300 4304 4305 4306 4307
0000320 4308 4309 4310 4311
0000340 4312 4313 4314 4315
0000360 4316 4317 4318 4319
0000400 4416 4417 4418 4419
0000420 4420 4421 4422 4423
0000440 4424 4425 4426 4427
0000460 4428 4429 4430 4431
```

Back, SIZE = 8192

```
0017300 16080 16081 16082 16083
0017320 16084 16085 16086 16087
0017340 16088 16089 16090 16091
0017360 16092 16093 16094 16095
0017400 16192 16193 16194 16195
0017420 16196 16197 16198 16199
0017440 16200 16201 16202 16203
0017460 16204 16205 16206 16207
0017500 16208 16209 16210 16211
0017520 16212 16213 16214 16215
0017540 16216 16217 16218 16219
0017560 16220 16221 16222 16223
0017600 16320 16321 16322 16323
0017620 16324 16325 16326 16327
0017640 16328 16329 16330 16331
0017660 16332 16333 16334 16335
0017700 16336 16337 16338 16339
0017720 16340 16341 16342 16343
0017740 16344 16345 16346 16347
0017760 16348 16349 16350 16351
0020000
[s22011631@sce 4by4]$ du -b SM7
8192 SM7
```

SM7 - 4by4 Partitioning

Front

```
[s22011631@sce 4by4]$ od -i SM8
0000000 4192 4193 4194 4195
0000020 4196 4197 4198 4199
0000040 4200 4201 4202 4203
0000060 4204 4205 4206 4207
0000100 4208 4209 4210 4211
0000120 4212 4213 4214 4215
0000140 4216 4217 4218 4219
0000160 4220 4221 4222 4223
0000200 4320 4321 4322 4323
0000220 4324 4325 4326 4327
0000240 4328 4329 4330 4331
0000260 4332 4333 4334 4335
0000300 4336 4337 4338 4339
0000320 4340 4341 4342 4343
0000340 4344 4345 4346 4347
0000360 4348 4349 4350 4351
0000400 4448 4449 4450 4451
0000420 4452 4453 4454 4455
0000440 4456 4457 4458 4459
0000460 4460 4461 4462 4463
```

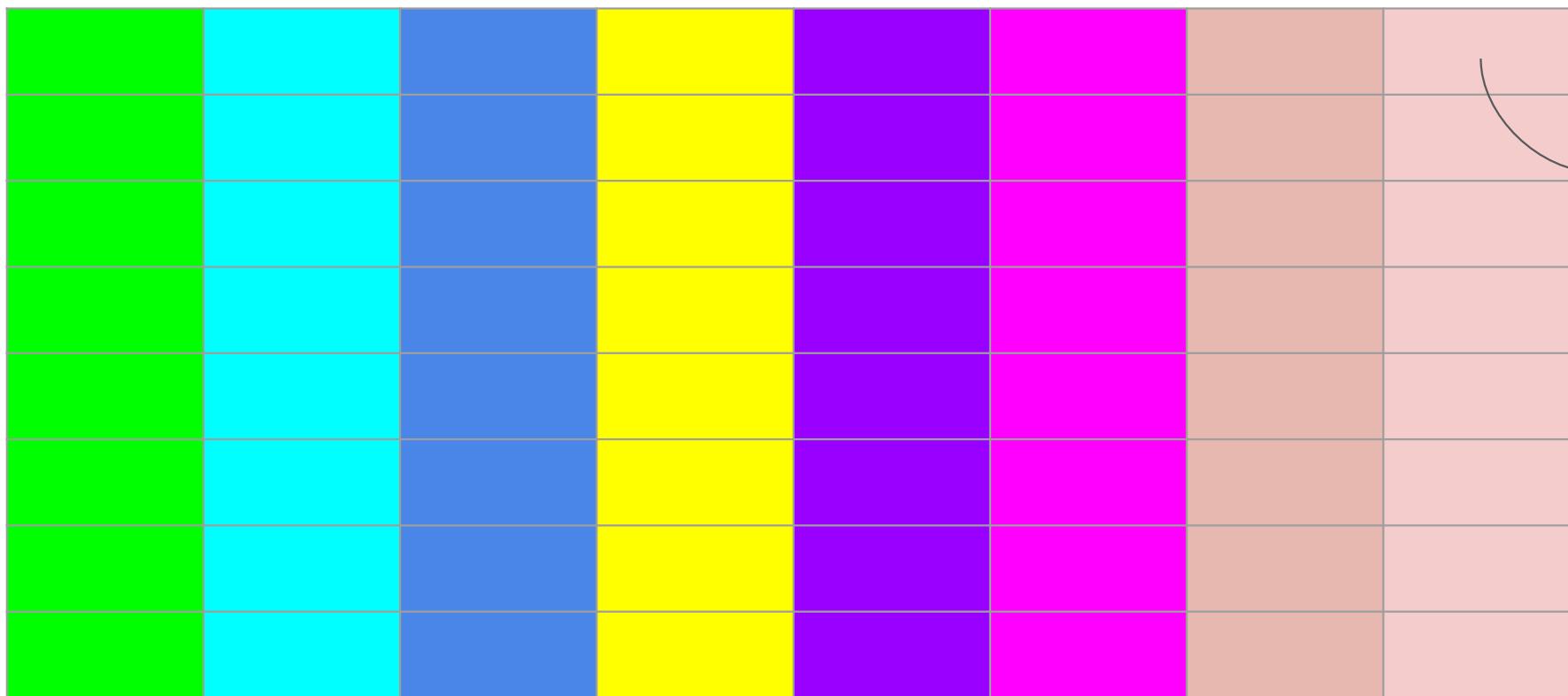
Back, SIZE = 8192

```
0017300 16112 16113 16114 16115
0017320 16116 16117 16118 16119
0017340 16120 16121 16122 16123
0017360 16124 16125 16126 16127
0017400 16224 16225 16226 16227
0017420 16228 16229 16230 16231
0017440 16232 16233 16234 16235
0017460 16236 16237 16238 16239
0017500 16240 16241 16242 16243
0017520 16244 16245 16246 16247
0017540 16248 16249 16250 16251
0017560 16252 16253 16254 16255
0017600 16352 16353 16354 16355
0017620 16356 16357 16358 16359
0017640 16360 16361 16362 16363
0017660 16364 16365 16366 16367
0017700 16368 16369 16370 16371
0017720 16372 16373 16374 16375
0017740 16376 16377 16378 16379
0017760 16380 16381 16382 16383
0020000
[s22011631@sce 4by4]$ du -b SM8
8192 SM8
```

8by8 Partitioning

128 x 128 정수 배열

SM0 SM1 SM2 SM3 SM4 SM5 SM6 SM7



블럭

8by8 Partitioning

```
// 각 SM에 해당하는 블록 쓰기
for (block_x = 0; block_x < NUM_BLOCKS; block_x++) {
    for (block_y = 0; block_y < NUM_BLOCKS; block_y++) {
        int block[BLOCK_SIZE][BLOCK_SIZE];

        for (i = 0; i < BLOCK_SIZE; i++) {
            for (j = 0; j < BLOCK_SIZE; j++) {
                block[i][j] = matrix[block_x * BLOCK_SIZE + i][block_y * BLOCK_SIZE + j];
            }
        }

        if (write(sm_fds[block_y], block, sizeof(block)) != sizeof(block)) {
            perror("SM 파일 쓰기 실패");
            return 1;
        }
    }
}
```

SM0 - 8by8 Partitioning

Front

```
[s22011631@sce 8by8]$ od -i SM1
0000000    0      1      2      3
0000020    4      5      6      7
0000040    8      9     10     11
0000060   12     13     14     15
0000100  128    129    130    131
0000120  132    133    134    135
0000140  136    137    138    139
0000160  140    141    142    143
0000200  256    257    258    259
0000220  260    261    262    263
0000240  264    265    266    267
0000260  268    269    270    271
0000300  384    385    386    387
0000320  388    389    390    391
0000340  392    393    394    395
0000360  396    397    398    399
0000400  512    513    514    515
0000420  516    517    518    519
0000440  520    521    522    523
0000460  524    525    526    527
```

Back, SIZE = 8192

```
0017300    15744    15745    15746    15747
0017320    15748    15749    15750    15751
0017340    15752    15753    15754    15755
0017360    15756    15757    15758    15759
0017400    15872    15873    15874    15875
0017420    15876    15877    15878    15879
0017440    15880    15881    15882    15883
0017460    15884    15885    15886    15887
0017500    16000    16001    16002    16003
0017520    16004    16005    16006    16007
0017540    16008    16009    16010    16011
0017560    16012    16013    16014    16015
0017600    16128    16129    16130    16131
0017620    16132    16133    16134    16135
0017640    16136    16137    16138    16139
0017660    16140    16141    16142    16143
0017700    16256    16257    16258    16259
0017720    16260    16261    16262    16263
0017740    16264    16265    16266    16267
0017760    16268    16269    16270    16271
0020000
[s22011631@sce 8by8]$ du -b SM1
8192    SM1
```

SM1 - 8by8 Partitioning

Front

```
[s22011631@sce 8by8]$ od -i SM2
0000000 16 17 18 19
0000020 20 21 22 23
0000040 24 25 26 27
0000060 28 29 30 31
0000100 144 145 146 147
0000120 148 149 150 151
0000140 152 153 154 155
0000160 156 157 158 159
0000200 272 273 274 275
0000220 276 277 278 279
0000240 280 281 282 283
0000260 284 285 286 287
0000300 400 401 402 403
0000320 404 405 406 407
0000340 408 409 410 411
0000360 412 413 414 415
0000400 528 529 530 531
0000420 532 533 534 535
0000440 536 537 538 539
0000460 540 541 542 543
```

Back, SIZE = 8192

```
0017300 15760 15761 15762 15763
0017320 15764 15765 15766 15767
0017340 15768 15769 15770 15771
0017360 15772 15773 15774 15775
0017400 15888 15889 15890 15891
0017420 15892 15893 15894 15895
0017440 15896 15897 15898 15899
0017460 15900 15901 15902 15903
0017500 16016 16017 16018 16019
0017520 16020 16021 16022 16023
0017540 16024 16025 16026 16027
0017560 16028 16029 16030 16031
0017600 16144 16145 16146 16147
0017620 16148 16149 16150 16151
0017640 16152 16153 16154 16155
0017660 16156 16157 16158 16159
0017700 16272 16273 16274 16275
0017720 16276 16277 16278 16279
0017740 16280 16281 16282 16283
0017760 16284 16285 16286 16287
0020000
[s22011631@sce 8by8]$ du -b SM2
8192 SM2
```

SM2 - 8by8 Partitioning

Front

```
[s22011631@sce 8by8]$ od -i SM3
0000000      32      33      34      35
00000020     36      37      38      39
00000040     40      41      42      43
00000060     44      45      46      47
0000100    160     161     162     163
0000120    164     165     166     167
0000140    168     169     170     171
0000160    172     173     174     175
0000200    288     289     290     291
0000220    292     293     294     295
0000240    296     297     298     299
0000260    300     301     302     303
0000300    416     417     418     419
0000320    420     421     422     423
0000340    424     425     426     427
0000360    428     429     430     431
0000400    544     545     546     547
0000420    548     549     550     551
0000440    552     553     554     555
0000460    556     557     558     559
```

Back, SIZE = 8192

0017300	15776	15777	15778	15779
0017320	15780	15781	15782	15783
0017340	15784	15785	15786	15787
0017360	15788	15789	15790	15791
0017400	15904	15905	15906	15907
0017420	15908	15909	15910	15911
0017440	15912	15913	15914	15915
0017460	15916	15917	15918	15919
0017500	16032	16033	16034	16035
0017520	16036	16037	16038	16039
0017540	16040	16041	16042	16043
0017560	16044	16045	16046	16047
0017600	16160	16161	16162	16163
0017620	16164	16165	16166	16167
0017640	16168	16169	16170	16171
0017660	16172	16173	16174	16175
0017700	16288	16289	16290	16291
0017720	16292	16293	16294	16295
0017740	16296	16297	16298	16299
0017760	16300	16301	16302	16303
0020000				

```
[s22011631@sce 8by8]$ du -b SM3
8192    SM3
```

SM3 - 8by8 Partitioning

Front

```
[s22011631@sce 8by8]$ od -i SM4
0000000    48      49      50      51
00000020   52      53      54      55
00000040   56      57      58      59
00000060   60      61      62      63
0000100   176     177     178     179
0000120   180     181     182     183
0000140   184     185     186     187
0000160   188     189     190     191
0000200   304     305     306     307
0000220   308     309     310     311
0000240   312     313     314     315
0000260   316     317     318     319
0000300   432     433     434     435
0000320   436     437     438     439
0000340   440     441     442     443
0000360   444     445     446     447
0000400   560     561     562     563
0000420   564     565     566     567
0000440   568     569     570     571
0000460   572     573     574     575
```

Back, SIZE = 8192

```
0017300    15792    15793    15794    15795
0017320    15796    15797    15798    15799
0017340    15800    15801    15802    15803
0017360    15804    15805    15806    15807
0017400    15920    15921    15922    15923
0017420    15924    15925    15926    15927
0017440    15928    15929    15930    15931
0017460    15932    15933    15934    15935
0017500    16048    16049    16050    16051
0017520    16052    16053    16054    16055
0017540    16056    16057    16058    16059
0017560    16060    16061    16062    16063
0017600    16176    16177    16178    16179
0017620    16180    16181    16182    16183
0017640    16184    16185    16186    16187
0017660    16188    16189    16190    16191
0017700    16304    16305    16306    16307
0017720    16308    16309    16310    16311
0017740    16312    16313    16314    16315
0017760    16316    16317    16318    16319
0020000
```

```
[s22011631@sce 8by8]$ du -b SM4
8192    SM4
```

SM4 - 8by8 Partitioning

Front

```
[s22011631@sce 8by8]$ od -i SM5
0000000    64      65      66      67
00000020   68      69      70      71
00000040   72      73      74      75
00000060   76      77      78      79
00000100  192     193     194     195
00000120  196     197     198     199
00000140  200     201     202     203
00000160  204     205     206     207
00000200  320     321     322     323
00000220  324     325     326     327
00000240  328     329     330     331
00000260  332     333     334     335
00000300  448     449     450     451
00000320  452     453     454     455
00000340  456     457     458     459
00000360  460     461     462     463
00000400  576     577     578     579
00000420  580     581     582     583
00000440  584     585     586     587
00000460  588     589     590     591
```

Back, SIZE = 8192

0017300	15808	15809	15810	15811
0017320	15812	15813	15814	15815
0017340	15816	15817	15818	15819
0017360	15820	15821	15822	15823
0017400	15936	15937	15938	15939
0017420	15940	15941	15942	15943
0017440	15944	15945	15946	15947
0017460	15948	15949	15950	15951
0017500	16064	16065	16066	16067
0017520	16068	16069	16070	16071
0017540	16072	16073	16074	16075
0017560	16076	16077	16078	16079
0017600	16192	16193	16194	16195
0017620	16196	16197	16198	16199
0017640	16200	16201	16202	16203
0017660	16204	16205	16206	16207
0017700	16320	16321	16322	16323
0017720	16324	16325	16326	16327
0017740	16328	16329	16330	16331
0017760	16332	16333	16334	16335
0020000				

```
[s22011631@sce 8by8]$ du -b SM5
8192    SM5
```

SM5 - 8by8 Partitioning

Front

```
[s22011631@sce 8by8]$ od -i SM6
0000000    80      81      82      83
0000020    84      85      86      87
0000040    88      89      90      91
0000060    92      93      94      95
0000100   208     209     210     211
0000120   212     213     214     215
0000140   216     217     218     219
0000160   220     221     222     223
0000200   336     337     338     339
0000220   340     341     342     343
0000240   344     345     346     347
0000260   348     349     350     351
0000300   464     465     466     467
0000320   468     469     470     471
0000340   472     473     474     475
0000360   476     477     478     479
0000400   592     593     594     595
0000420   596     597     598     599
0000440   600     601     602     603
0000460   604     605     606     607
```

Back, SIZE = 8192

```
0017300    15824    15825    15826    15827
0017320    15828    15829    15830    15831
0017340    15832    15833    15834    15835
0017360    15836    15837    15838    15839
0017400    15952    15953    15954    15955
0017420    15956    15957    15958    15959
0017440    15960    15961    15962    15963
0017460    15964    15965    15966    15967
0017500    16080    16081    16082    16083
0017520    16084    16085    16086    16087
0017540    16088    16089    16090    16091
0017560    16092    16093    16094    16095
0017600    16208    16209    16210    16211
0017620    16212    16213    16214    16215
0017640    16216    16217    16218    16219
0017660    16220    16221    16222    16223
0017700    16336    16337    16338    16339
0017720    16340    16341    16342    16343
0017740    16344    16345    16346    16347
0017760    16348    16349    16350    16351
0020000
[s22011631@sce 8by8]$ du -b SM6
8192      SM6
```

SM6 - 8by8 Partitioning

Front

```
[s22011631@sce 8by8]$ od -i SM7
0000000      96      97      98      99
00000020     100     101     102     103
00000040     104     105     106     107
00000060     108     109     110     111
00000100     224     225     226     227
00000120     228     229     230     231
00000140     232     233     234     235
00000160     236     237     238     239
00000200     352     353     354     355
00000220     356     357     358     359
00000240     360     361     362     363
00000260     364     365     366     367
00000300     480     481     482     483
00000320     484     485     486     487
00000340     488     489     490     491
00000360     492     493     494     495
00000400     608     609     610     611
00000420     612     613     614     615
00000440     616     617     618     619
00000460     620     621     622     623
```

Back, SIZE = 8192

```
0017300      15840    15841    15842    15843
0017320      15844    15845    15846    15847
0017340      15848    15849    15850    15851
0017360      15852    15853    15854    15855
0017400      15968    15969    15970    15971
0017420      15972    15973    15974    15975
0017440      15976    15977    15978    15979
0017460      15980    15981    15982    15983
0017500      16096    16097    16098    16099
0017520      16100    16101    16102    16103
0017540      16104    16105    16106    16107
0017560      16108    16109    16110    16111
0017600      16224    16225    16226    16227
0017620      16228    16229    16230    16231
0017640      16232    16233    16234    16235
0017660      16236    16237    16238    16239
0017700      16352    16353    16354    16355
0017720      16356    16357    16358    16359
0017740      16360    16361    16362    16363
0017760      16364    16365    16366    16367
0020000
[s22011631@sce 8by8]$ du -b SM7
8192      SM7
```

SM7 - 8by8 Partitioning

Front

```
[s22011631@sce 8by8]$ od -i SM8
0000000    112    113    114    115
0000020    116    117    118    119
0000040    120    121    122    123
0000060    124    125    126    127
0000100    240    241    242    243
0000120    244    245    246    247
0000140    248    249    250    251
0000160    252    253    254    255
0000200    368    369    370    371
0000220    372    373    374    375
0000240    376    377    378    379
0000260    380    381    382    383
0000300    496    497    498    499
0000320    500    501    502    503
0000340    504    505    506    507
0000360    508    509    510    511
0000400    624    625    626    627
0000420    628    629    630    631
0000440    632    633    634    635
0000460    636    637    638    639
```

Back, SIZE = 8192

```
0017300    15856    15857    15858    15859
0017320    15860    15861    15862    15863
0017340    15864    15865    15866    15867
0017360    15868    15869    15870    15871
0017400    15984    15985    15986    15987
0017420    15988    15989    15990    15991
0017440    15992    15993    15994    15995
0017460    15996    15997    15998    15999
0017500    16112    16113    16114    16115
0017520    16116    16117    16118    16119
0017540    16120    16121    16122    16123
0017560    16124    16125    16126    16127
0017600    16240    16241    16242    16243
0017620    16244    16245    16246    16247
0017640    16248    16249    16250    16251
0017660    16252    16253    16254    16255
0017700    16368    16369    16370    16371
0017720    16372    16373    16374    16375
0017740    16376    16377    16378    16379
0017760    16380    16381    16382    16383
0020000
```

```
[s22011631@sce 8by8]$ du -b SM8
8192    SM8
```

Client-Client Communication

8 by 8

client - client , 8x8 Partitioning

전체 SM_i File Size : 8192 (각 SM File 당 2048개의 정수)

-rw-r--r--.	1	s20011914	class	8192	2024-12-04	18:42	SM_1
-rw-r--r--.	1	s20011914	class	8192	2024-12-04	18:42	SM_2
-rw-r--r--.	1	s20011914	class	8192	2024-12-04	18:42	SM_3
-rw-r--r--.	1	s20011914	class	8192	2024-12-04	18:42	SM_4
-rw-r--r--.	1	s20011914	class	8192	2024-12-04	18:42	SM_5
-rw-r--r--.	1	s20011914	class	8192	2024-12-04	18:42	SM_6
-rw-r--r--.	1	s20011914	class	8192	2024-12-04	18:42	SM_7
-rw-r--r--.	1	s20011914	class	8192	2024-12-04	18:42	SM_8

```
[s20011914@sce 1204]$ od -i SM_1
0000000    0      1      2      3
0000020    4      5      6      7
0000040    8      9     10     11
0000060   12     13     14     15
0000100   16     17     18     19
0000120   20     21     22     23
0000140   24     25     26     27
0000160   28     29     30     31
0000200   32     33     34     35
0000220   36     37     38     39
0000240   40     41     42     43
0000260   44     45     46     47
0000300   48     49     50     51
0000320   52     53     54     55
0000340   56     57     58     59
0000360   60     61     62     63
0000400   64     65     66     67
0000420   68     69     70     71
0000440   72     73     74     75
0000460   76     77     78     79
0000500   80     81     82     83
0000520   84     85     86     87
0000540   88     89     90     91
0000560   92     93     94     95
0000600   96     97     98     99
0000620  100    101    102    103
0000640  104    105    106    107
0000660  108    109    110    111
0000700  112    113    114    115
0000720  116    117    118    119
0000740  120    121    122    123
0000760  124    125    126    127
0001000  128    129    130    131
0001020  132    133    134    135
0001040  136    137    138    139
0001060  140    141    142    143
0001100  144    145    146    147
0001120  148    149    150    151
0001140  152    153    154    155
0001160  156    157    158    159
0001200  160    161    162    163
0001220  164    165    166    167
0001240  168    169    170    171
0001260  172    173    174    175
```

SM_0 Front

0016520	1876	1877	1878	1879
0016540	1880	1881	1882	1883
0016560	1884	1885	1886	1887
0016600	1888	1889	1890	1891
0016620	1892	1893	1894	1895
0016640	1896	1897	1898	1899
0016660	1900	1901	1902	1903
0016700	1904	1905	1906	1907
0016720	1908	1909	1910	1911
0016740	1912	1913	1914	1915
0016760	1916	1917	1918	1919
0017000	1920	1921	1922	1923
0017020	1924	1925	1926	1927
0017040	1928	1929	1930	1931
0017060	1932	1933	1934	1935
0017100	1936	1937	1938	1939
0017120	1940	1941	1942	1943
0017140	1944	1945	1946	1947
0017160	1948	1949	1950	1951
0017200	1952	1953	1954	1955
0017220	1956	1957	1958	1959
0017240	1960	1961	1962	1963
0017260	1964	1965	1966	1967
0017300	1968	1969	1970	1971
0017320	1972	1973	1974	1975
0017340	1976	1977	1978	1979
0017360	1980	1981	1982	1983
0017400	1984	1985	1986	1987
0017420	1988	1989	1990	1991
0017440	1992	1993	1994	1995
0017460	1996	1997	1998	1999
0017500	2000	2001	2002	2003
0017520	2004	2005	2006	2007
0017540	2008	2009	2010	2011
0017560	2012	2013	2014	2015
0017600	2016	2017	2018	2019
0017620	2020	2021	2022	2023
0017640	2024	2025	2026	2027
0017660	2028	2029	2030	2031
0017700	2032	2033	2034	2035
0017720	2036	2037	2038	2039
0017740	2040	2041	2042	2043
0017760	2044	2045	2046	2047
0020000				

[s20011914@sce 1204]\$

SM_0 back

```
[s20011914@sce 1204]$ od -i SM_2
0000000 2048 2049 2050 2051
0000020 2052 2053 2054 2055
0000040 2056 2057 2058 2059
0000060 2060 2061 2062 2063
0000100 2064 2065 2066 2067
0000120 2068 2069 2070 2071
0000140 2072 2073 2074 2075
0000160 2076 2077 2078 2079
0000200 2080 2081 2082 2083
0000220 2084 2085 2086 2087
0000240 2088 2089 2090 2091
0000260 2092 2093 2094 2095
0000300 2096 2097 2098 2099
0000320 2100 2101 2102 2103
0000340 2104 2105 2106 2107
0000360 2108 2109 2110 2111
0000400 2112 2113 2114 2115
0000420 2116 2117 2118 2119
0000440 2120 2121 2122 2123
0000460 2124 2125 2126 2127
0000500 2128 2129 2130 2131
0000520 2132 2133 2134 2135
0000540 2136 2137 2138 2139
0000560 2140 2141 2142 2143
0000600 2144 2145 2146 2147
0000620 2148 2149 2150 2151
0000640 2152 2153 2154 2155
0000660 2156 2157 2158 2159
0000700 2160 2161 2162 2163
0000720 2164 2165 2166 2167
0000740 2168 2169 2170 2171
0000760 2172 2173 2174 2175
0001000 2176 2177 2178 2179
0001020 2180 2181 2182 2183
0001040 2184 2185 2186 2187
0001060 2188 2189 2190 2191
0001100 2192 2193 2194 2195
0001120 2196 2197 2198 2199
0001140 2200 2201 2202 2203
0001160 2204 2205 2206 2207
0001200 2208 2209 2210 2211
0001220 2212 2213 2214 2215
0001240 2216 2217 2218 2219
0001260 2220 2221 2222 2223
```

SM_1 Front

```
0016520 3924 3925 3926 3927
0016540 3928 3929 3930 3931
0016560 3932 3933 3934 3935
0016600 3936 3937 3938 3939
0016620 3940 3941 3942 3943
0016640 3944 3945 3946 3947
0016660 3948 3949 3950 3951
0016700 3952 3953 3954 3955
0016720 3956 3957 3958 3959
0016740 3960 3961 3962 3963
0016760 3964 3965 3966 3967
0017000 3968 3969 3970 3971
0017020 3972 3973 3974 3975
0017040 3976 3977 3978 3979
0017060 3980 3981 3982 3983
0017100 3984 3985 3986 3987
0017120 3988 3989 3990 3991
0017140 3992 3993 3994 3995
0017160 3996 3997 3998 3999
0017200 4000 4001 4002 4003
0017220 4004 4005 4006 4007
0017240 4008 4009 4010 4011
0017260 4012 4013 4014 4015
0017300 4016 4017 4018 4019
0017320 4020 4021 4022 4023
0017340 4024 4025 4026 4027
0017360 4028 4029 4030 4031
0017400 4032 4033 4034 4035
0017420 4036 4037 4038 4039
0017440 4040 4041 4042 4043
0017460 4044 4045 4046 4047
0017500 4048 4049 4050 4051
0017520 4052 4053 4054 4055
0017540 4056 4057 4058 4059
0017560 4060 4061 4062 4063
0017600 4064 4065 4066 4067
0017620 4068 4069 4070 4071
0017640 4072 4073 4074 4075
0017660 4076 4077 4078 4079
0017700 4080 4081 4082 4083
0017720 4084 4085 4086 4087
0017740 4088 4089 4090 4091
0017760 4092 4093 4094 4095
0020000
[s20011914@sce 1204]$
```

SM_1 back

```
[s20011914@sce 1204]$ od -i SM_3
0000000 4096 4097 4098 4099
0000020 4100 4101 4102 4103
0000040 4104 4105 4106 4107
0000060 4108 4109 4110 4111
0000100 4112 4113 4114 4115
0000120 4116 4117 4118 4119
0000140 4120 4121 4122 4123
0000160 4124 4125 4126 4127
0000200 4128 4129 4130 4131
0000220 4132 4133 4134 4135
0000240 4136 4137 4138 4139
0000260 4140 4141 4142 4143
0000300 4144 4145 4146 4147
0000320 4148 4149 4150 4151
0000340 4152 4153 4154 4155
0000360 4156 4157 4158 4159
0000400 4160 4161 4162 4163
0000420 4164 4165 4166 4167
0000440 4168 4169 4170 4171
0000460 4172 4173 4174 4175
0000500 4176 4177 4178 4179
0000520 4180 4181 4182 4183
0000540 4184 4185 4186 4187
0000560 4188 4189 4190 4191
0000600 4192 4193 4194 4195
0000620 4196 4197 4198 4199
0000640 4200 4201 4202 4203
0000660 4204 4205 4206 4207
0000700 4208 4209 4210 4211
0000720 4212 4213 4214 4215
0000740 4216 4217 4218 4219
0000760 4220 4221 4222 4223
0001000 4224 4225 4226 4227
0001020 4228 4229 4230 4231
0001040 4232 4233 4234 4235
0001060 4236 4237 4238 4239
0001100 4240 4241 4242 4243
0001120 4244 4245 4246 4247
0001140 4248 4249 4250 4251
0001160 4252 4253 4254 4255
0001200 4256 4257 4258 4259
0001220 4260 4261 4262 4263
0001240 4264 4265 4266 4267
```

SM_2 Front

```
0016520 5972 5973 5974 5975
0016540 5976 5977 5978 5979
0016560 5980 5981 5982 5983
0016600 5984 5985 5986 5987
0016620 5988 5989 5990 5991
0016640 5992 5993 5994 5995
0016660 5996 5997 5998 5999
0016700 6000 6001 6002 6003
0016720 6004 6005 6006 6007
0016740 6008 6009 6010 6011
0016760 6012 6013 6014 6015
0017000 6016 6017 6018 6019
0017020 6020 6021 6022 6023
0017040 6024 6025 6026 6027
0017060 6028 6029 6030 6031
0017100 6032 6033 6034 6035
0017120 6036 6037 6038 6039
0017140 6040 6041 6042 6043
0017160 6044 6045 6046 6047
0017200 6048 6049 6050 6051
0017220 6052 6053 6054 6055
0017240 6056 6057 6058 6059
0017260 6060 6061 6062 6063
0017300 6064 6065 6066 6067
0017320 6068 6069 6070 6071
0017340 6072 6073 6074 6075
0017360 6076 6077 6078 6079
0017400 6080 6081 6082 6083
0017420 6084 6085 6086 6087
0017440 6088 6089 6090 6091
0017460 6092 6093 6094 6095
0017500 6096 6097 6098 6099
0017520 6100 6101 6102 6103
0017540 6104 6105 6106 6107
0017560 6108 6109 6110 6111
0017600 6112 6113 6114 6115
0017620 6116 6117 6118 6119
0017640 6120 6121 6122 6123
0017660 6124 6125 6126 6127
0017700 6128 6129 6130 6131
0017720 6132 6133 6134 6135
0017740 6136 6137 6138 6139
0017760 6140 6141 6142 6143
0020000
```

SM_2 back

```
[s20011914@sce 1204]$ cd 1204
```

```
[s20011914@sce 1204]$ od -i SM_4
0000000 6144 6145 6146 6147
0000020 6148 6149 6150 6151
0000040 6152 6153 6154 6155
0000060 6156 6157 6158 6159
0000100 6160 6161 6162 6163
0000120 6164 6165 6166 6167
0000140 6168 6169 6170 6171
0000160 6172 6173 6174 6175
0000200 6176 6177 6178 6179
0000220 6180 6181 6182 6183
0000240 6184 6185 6186 6187
0000260 6188 6189 6190 6191
0000300 6192 6193 6194 6195
0000320 6196 6197 6198 6199
0000340 6200 6201 6202 6203
0000360 6204 6205 6206 6207
0000400 6208 6209 6210 6211
0000420 6212 6213 6214 6215
0000440 6216 6217 6218 6219
0000460 6220 6221 6222 6223
0000500 6224 6225 6226 6227
0000520 6228 6229 6230 6231
0000540 6232 6233 6234 6235
0000560 6236 6237 6238 6239
0000600 6240 6241 6242 6243
0000620 6244 6245 6246 6247
0000640 6248 6249 6250 6251
0000660 6252 6253 6254 6255
0000700 6256 6257 6258 6259
0000720 6260 6261 6262 6263
0000740 6264 6265 6266 6267
0000760 6268 6269 6270 6271
0001000 6272 6273 6274 6275
0001020 6276 6277 6278 6279
0001040 6280 6281 6282 6283
0001060 6284 6285 6286 6287
0001100 6288 6289 6290 6291
0001120 6292 6293 6294 6295
0001140 6296 6297 6298 6299
0001160 6300 6301 6302 6303
0001200 6304 6305 6306 6307
0001220 6308 6309 6310 6311
0001240 6312 6313 6314 6315
0001260 6316 6317 6318 6319
```

SM_3 Front

```
0016520 8020 8021 8022 8023
0016540 8024 8025 8026 8027
0016560 8028 8029 8030 8031
0016600 8032 8033 8034 8035
0016620 8036 8037 8038 8039
0016640 8040 8041 8042 8043
0016660 8044 8045 8046 8047
0016700 8048 8049 8050 8051
0016720 8052 8053 8054 8055
0016740 8056 8057 8058 8059
0016760 8060 8061 8062 8063
0017000 8064 8065 8066 8067
0017020 8068 8069 8070 8071
0017040 8072 8073 8074 8075
0017060 8076 8077 8078 8079
0017100 8080 8081 8082 8083
0017120 8084 8085 8086 8087
0017140 8088 8089 8090 8091
0017160 8092 8093 8094 8095
0017200 8096 8097 8098 8099
0017220 8100 8101 8102 8103
0017240 8104 8105 8106 8107
0017260 8108 8109 8110 8111
0017300 8112 8113 8114 8115
0017320 8116 8117 8118 8119
0017340 8120 8121 8122 8123
0017360 8124 8125 8126 8127
0017400 8128 8129 8130 8131
0017420 8132 8133 8134 8135
0017440 8136 8137 8138 8139
0017460 8140 8141 8142 8143
0017500 8144 8145 8146 8147
0017520 8148 8149 8150 8151
0017540 8152 8153 8154 8155
0017560 8156 8157 8158 8159
0017600 8160 8161 8162 8163
0017620 8164 8165 8166 8167
0017640 8168 8169 8170 8171
0017660 8172 8173 8174 8175
0017700 8176 8177 8178 8179
0017720 8180 8181 8182 8183
0017740 8184 8185 8186 8187
0017760 8188 8189 8190 8191
0020000
```

SM_3 back

```
[s20011914@sce 1204]$ od -i SM_5
0000000 8192 8193 8194 8195
0000020 8196 8197 8198 8199
0000040 8200 8201 8202 8203
0000060 8204 8205 8206 8207
0000100 8208 8209 8210 8211
0000120 8212 8213 8214 8215
0000140 8216 8217 8218 8219
0000160 8220 8221 8222 8223
0000200 8224 8225 8226 8227
0000220 8228 8229 8230 8231
0000240 8232 8233 8234 8235
0000260 8236 8237 8238 8239
0000300 8240 8241 8242 8243
0000320 8244 8245 8246 8247
0000340 8248 8249 8250 8251
0000360 8252 8253 8254 8255
0000400 8256 8257 8258 8259
0000420 8260 8261 8262 8263
0000440 8264 8265 8266 8267
0000460 8268 8269 8270 8271
0000500 8272 8273 8274 8275
0000520 8276 8277 8278 8279
0000540 8280 8281 8282 8283
0000560 8284 8285 8286 8287
0000600 8288 8289 8290 8291
0000620 8292 8293 8294 8295
0000640 8296 8297 8298 8299
0000660 8300 8301 8302 8303
0000700 8304 8305 8306 8307
0000720 8308 8309 8310 8311
0000740 8312 8313 8314 8315
0000760 8316 8317 8318 8319
0001000 8320 8321 8322 8323
0001020 8324 8325 8326 8327
0001040 8328 8329 8330 8331
0001060 8332 8333 8334 8335
0001100 8336 8337 8338 8339
0001120 8340 8341 8342 8343
0001140 8344 8345 8346 8347
0001160 8348 8349 8350 8351
0001200 8352 8353 8354 8355
0001220 8356 8357 8358 8359
0001240 8360 8361 8362 8363
0001260 8364 8365 8366 8367
```

SM_4 Front

```
0016520 10068 10069 10070 10071
0016540 10072 10073 10074 10075
0016560 10076 10077 10078 10079
0016600 10080 10081 10082 10083
0016620 10084 10085 10086 10087
0016640 10088 10089 10090 10091
0016660 10092 10093 10094 10095
0016700 10096 10097 10098 10099
0016720 10100 10101 10102 10103
0016740 10104 10105 10106 10107
0016760 10108 10109 10110 10111
0017000 10112 10113 10114 10115
0017020 10116 10117 10118 10119
0017040 10120 10121 10122 10123
0017060 10124 10125 10126 10127
0017100 10128 10129 10130 10131
0017120 10132 10133 10134 10135
0017140 10136 10137 10138 10139
0017160 10140 10141 10142 10143
0017200 10144 10145 10146 10147
0017220 10148 10149 10150 10151
0017240 10152 10153 10154 10155
0017260 10156 10157 10158 10159
0017300 10160 10161 10162 10163
0017320 10164 10165 10166 10167
0017340 10168 10169 10170 10171
0017360 10172 10173 10174 10175
0017400 10176 10177 10178 10179
0017420 10180 10181 10182 10183
0017440 10184 10185 10186 10187
0017460 10188 10189 10190 10191
0017500 10192 10193 10194 10195
0017520 10196 10197 10198 10199
0017540 10200 10201 10202 10203
0017560 10204 10205 10206 10207
0017600 10208 10209 10210 10211
0017620 10212 10213 10214 10215
0017640 10216 10217 10218 10219
0017660 10220 10221 10222 10223
0017700 10224 10225 10226 10227
0017720 10228 10229 10230 10231
0017740 10232 10233 10234 10235
0017760 10236 10237 10238 10239
0020000
```

SM_4 back

```
[s20011914@sce 1204]$ od -i SM_6
0000000 10240 10241 10242 10243
0000020 10244 10245 10246 10247
0000040 10248 10249 10250 10251
0000060 10252 10253 10254 10255
0000100 10256 10257 10258 10259
0000120 10260 10261 10262 10263
0000140 10264 10265 10266 10267
0000160 10268 10269 10270 10271
0000200 10272 10273 10274 10275
0000220 10276 10277 10278 10279
0000240 10280 10281 10282 10283
0000260 10284 10285 10286 10287
0000300 10288 10289 10290 10291
0000320 10292 10293 10294 10295
0000340 10296 10297 10298 10299
0000360 10300 10301 10302 10303
0000400 10304 10305 10306 10307
0000420 10308 10309 10310 10311
0000440 10312 10313 10314 10315
0000460 10316 10317 10318 10319
0000500 10320 10321 10322 10323
0000520 10324 10325 10326 10327
0000540 10328 10329 10330 10331
0000560 10332 10333 10334 10335
0000600 10336 10337 10338 10339
0000620 10340 10341 10342 10343
0000640 10344 10345 10346 10347
0000660 10348 10349 10350 10351
0000700 10352 10353 10354 10355
0000720 10356 10357 10358 10359
0000740 10360 10361 10362 10363
0000760 10364 10365 10366 10367
0001000 10368 10369 10370 10371
0001020 10372 10373 10374 10375
0001040 10376 10377 10378 10379
0001060 10380 10381 10382 10383
0001100 10384 10385 10386 10387
0001120 10388 10389 10390 10391
0001140 10392 10393 10394 10395
0001160 10396 10397 10398 10399
0001200 10400 10401 10402 10403
0001220 10404 10405 10406 10407
0001240 10408 10409 10410 10411
0001260 10412 10413 10414 10415
```

SM_5 Front

```
0016520 12116 12117 12118 12119
0016540 12120 12121 12122 12123
0016560 12124 12125 12126 12127
0016600 12128 12129 12130 12131
0016620 12132 12133 12134 12135
0016640 12136 12137 12138 12139
0016660 12140 12141 12142 12143
0016700 12144 12145 12146 12147
0016720 12148 12149 12150 12151
0016740 12152 12153 12154 12155
0016760 12156 12157 12158 12159
0017000 12160 12161 12162 12163
0017020 12164 12165 12166 12167
0017040 12168 12169 12170 12171
0017060 12172 12173 12174 12175
0017100 12176 12177 12178 12179
0017120 12180 12181 12182 12183
0017140 12184 12185 12186 12187
0017160 12188 12189 12190 12191
0017200 12192 12193 12194 12195
0017220 12196 12197 12198 12199
0017240 12200 12201 12202 12203
0017260 12204 12205 12206 12207
0017300 12208 12209 12210 12211
0017320 12212 12213 12214 12215
0017340 12216 12217 12218 12219
0017360 12220 12221 12222 12223
0017400 12224 12225 12226 12227
0017420 12228 12229 12230 12231
0017440 12232 12233 12234 12235
0017460 12236 12237 12238 12239
0017500 12240 12241 12242 12243
0017520 12244 12245 12246 12247
0017540 12248 12249 12250 12251
0017560 12252 12253 12254 12255
0017600 12256 12257 12258 12259
0017620 12260 12261 12262 12263
0017640 12264 12265 12266 12267
0017660 12268 12269 12270 12271
0017700 12272 12273 12274 12275
0017720 12276 12277 12278 12279
0017740 12280 12281 12282 12283
0017760 12284 12285 12286 12287
0020000
[s20011914@sce 1204]$ █
```

SM_5 back

```
[s20011914@sce 1204]$ od -i SM_7
0000000 12288 12289 12290 12291
0000020 12292 12293 12294 12295
0000040 12296 12297 12298 12299
0000060 12300 12301 12302 12303
0000100 12304 12305 12306 12307
0000120 12308 12309 12310 12311
0000140 12312 12313 12314 12315
0000160 12316 12317 12318 12319
0000200 12320 12321 12322 12323
0000220 12324 12325 12326 12327
0000240 12328 12329 12330 12331
0000260 12332 12333 12334 12335
0000300 12336 12337 12338 12339
0000320 12340 12341 12342 12343
0000340 12344 12345 12346 12347
0000360 12348 12349 12350 12351
0000400 12352 12353 12354 12355
0000420 12356 12357 12358 12359
0000440 12360 12361 12362 12363
0000460 12364 12365 12366 12367
0000500 12368 12369 12370 12371
0000520 12372 12373 12374 12375
0000540 12376 12377 12378 12379
0000560 12380 12381 12382 12383
0000600 12384 12385 12386 12387
0000620 12388 12389 12390 12391
0000640 12392 12393 12394 12395
0000660 12396 12397 12398 12399
0000700 12400 12401 12402 12403
0000720 12404 12405 12406 12407
0000740 12408 12409 12410 12411
0000760 12412 12413 12414 12415
0001000 12416 12417 12418 12419
0001020 12420 12421 12422 12423
0001040 12424 12425 12426 12427
0001060 12428 12429 12430 12431
0001100 12432 12433 12434 12435
0001120 12436 12437 12438 12439
0001140 12440 12441 12442 12443
0001160 12444 12445 12446 12447
0001200 12448 12449 12450 12451
0001220 12452 12453 12454 12455
0001240 12456 12457 12458 12459
0001260 12460 12461 12462 12463
```

SM_6 Front

```
0016520 14164 14165 14166 14167
0016540 14168 14169 14170 14171
0016560 14172 14173 14174 14175
0016600 14176 14177 14178 14179
0016620 14180 14181 14182 14183
0016640 14184 14185 14186 14187
0016660 14188 14189 14190 14191
0016700 14192 14193 14194 14195
0016720 14196 14197 14198 14199
0016740 14200 14201 14202 14203
0016760 14204 14205 14206 14207
0017000 14208 14209 14210 14211
0017020 14212 14213 14214 14215
0017040 14216 14217 14218 14219
0017060 14220 14221 14222 14223
0017100 14224 14225 14226 14227
0017120 14228 14229 14230 14231
0017140 14232 14233 14234 14235
0017160 14236 14237 14238 14239
0017200 14240 14241 14242 14243
0017220 14244 14245 14246 14247
0017240 14248 14249 14250 14251
0017260 14252 14253 14254 14255
0017300 14256 14257 14258 14259
0017320 14260 14261 14262 14263
0017340 14264 14265 14266 14267
0017360 14268 14269 14270 14271
0017400 14272 14273 14274 14275
0017420 14276 14277 14278 14279
0017440 14280 14281 14282 14283
0017460 14284 14285 14286 14287
0017500 14288 14289 14290 14291
0017520 14292 14293 14294 14295
0017540 14296 14297 14298 14299
0017560 14300 14301 14302 14303
0017600 14304 14305 14306 14307
0017620 14308 14309 14310 14311
0017640 14312 14313 14314 14315
0017660 14316 14317 14318 14319
0017700 14320 14321 14322 14323
0017720 14324 14325 14326 14327
0017740 14328 14329 14330 14331
0017760 14332 14333 14334 14335
0020000
[s20011914@sce 1204]$
```

SM_6 back

[s20011914@sce 1204]\$ od -i SM_8	0000000 14336 14337 14338 14339	0016520 16212 16213 16214 16215
0000020 14340 14341 14342 14343	0016540 16216 16217 16218 16219	
0000040 14344 14345 14346 14347	0016560 16220 16221 16222 16223	
0000060 14348 14349 14350 14351	0016600 16224 16225 16226 16227	
0000100 14352 14353 14354 14355	0016620 16228 16229 16230 16231	
0000120 14356 14357 14358 14359	0016640 16232 16233 16234 16235	
0000140 14360 14361 14362 14363	0016660 16236 16237 16238 16239	
0000160 14364 14365 14366 14367	0016700 16240 16241 16242 16243	
0000200 14368 14369 14370 14371	0016720 16244 16245 16246 16247	
0000220 14372 14373 14374 14375	0016740 16248 16249 16250 16251	
0000240 14376 14377 14378 14379	0016760 16252 16253 16254 16255	
0000260 14380 14381 14382 14383	0017000 16256 16257 16258 16259	
0000300 14384 14385 14386 14387	0017020 16260 16261 16262 16263	
0000320 14388 14389 14390 14391	0017040 16264 16265 16266 16267	
0000340 14392 14393 14394 14395	0017060 16268 16269 16270 16271	
0000360 14396 14397 14398 14399	0017100 16272 16273 16274 16275	
0000400 14400 14401 14402 14403	0017120 16276 16277 16278 16279	
0000420 14404 14405 14406 14407	0017140 16280 16281 16282 16283	
0000440 14408 14409 14410 14411	0017160 16284 16285 16286 16287	
0000460 14412 14413 14414 14415	0017200 16288 16289 16290 16291	
0000500 14416 14417 14418 14419	0017220 16292 16293 16294 16295	
0000520 14420 14421 14422 14423	0017240 16296 16297 16298 16299	
0000540 14424 14425 14426 14427	0017260 16300 16301 16302 16303	
0000560 14428 14429 14430 14431	0017300 16304 16305 16306 16307	
0000600 14432 14433 14434 14435	0017320 16308 16309 16310 16311	
0000620 14436 14437 14438 14439	0017340 16312 16313 16314 16315	
0000640 14440 14441 14442 14443	0017360 16316 16317 16318 16319	
0000660 14444 14445 14446 14447	0017400 16320 16321 16322 16323	
0000700 14448 14449 14450 14451	0017420 16324 16325 16326 16327	
0000720 14452 14453 14454 14455	0017440 16328 16329 16330 16331	
0000740 14456 14457 14458 14459	0017460 16332 16333 16334 16335	
0000760 14460 14461 14462 14463	0017500 16336 16337 16338 16339	
0001000 14464 14465 14466 14467	0017520 16340 16341 16342 16343	
0001020 14468 14469 14470 14471	0017540 16344 16345 16346 16347	
0001040 14472 14473 14474 14475	0017560 16348 16349 16350 16351	
0001060 14476 14477 14478 14479	0017600 16352 16353 16354 16355	
0001100 14480 14481 14482 14483	0017620 16356 16357 16358 16359	
0001120 14484 14485 14486 14487	0017640 16360 16361 16362 16363	
0001140 14488 14489 14490 14491	0017660 16364 16365 16366 16367	
0001160 14492 14493 14494 14495	0017700 16368 16369 16370 16371	
0001200 14496 14497 14498 14499	0017720 16372 16373 16374 16375	
0001220 14500 14501 14502 14503	0017740 16376 16377 16378 16379	
0001240 14504 14505 14506 14507	0017760 16380 16381 16382 16383	
0001260 14508 14509 14510 14511	0020000	

[s20011914@sce 1204]\$

SM_7 Front

SM_7 back

client - client , 8x8 Partitioning

```
#define BLOCK_SIZE 16  
#define SM_COUNT 8  
#define ROWS_PER_FILE 16  
#define MSG_COUNT 16  
#define MSG_KEY_BASE 1000
```

BLOCK SIZE 16 : 128 X 128 을 64개 블럭으로 나눌 시 한 블럭 크기 16 X 16

SM_COUNT 8

ROWS_PER_FILE 16 : 1 Block = 16 rows

MSG_COUNT 16 : 한 번에 16개 정수 이동

MSG_KEY_BASE 1000 : 키 생성 시 이용

client - client , 8x8 Partitioning

```
void error_exit(const char* msg) {  
    perror(msg);  
    exit(1);  
}
```

오류 발생 메세지 출력 함수

```
struct message {  
    long msg_type;  
    int data[BLOCK_SIZE];  
};
```

메세지 구조체 : 정수형 배열 송/수신

```
void close_(int* smfd, int* sm_fd, int* msgq_id) {  
    int i;  
    for (i = 0; i < SM_COUNT; ++i) {  
        close(smfd[i]);  
        close(sm_fd[i]);  
    }  
  
    for (i = 0; i < SM_COUNT; ++i) {  
        if (msgctl(msgq_id[i], IPC_RMID, NULL) == -1) {  
            perror("msgctl");  
        }  
    }  
}
```

메세지 큐 & 기존 SM 파일, 새로운 SM 파일 닫기

client - client , 8x8 Partitioning

main 함수- 프로세스 시간 측정

```
struct timeval start, end;
long seconds, microseconds;
double elapsed;

gettimeofday(&start, NULL);
```

```
gettimeofday(&end, NULL);

seconds = end.tv_sec - start.tv_sec;
microseconds = end.tv_usec - start.tv_usec;

elapsed = seconds + microseconds * 1e-6;

printf("process execute time : %.6f sec\n", elapsed);
```

송/수신 프로세스 시작 전 시각 측정

송/수신 프로세스 후 시각 측정

종료 시각 - 시작 시각 = 프로세스 실행 시간

client - client , 8x8 Partitioning

main 함수 - 메세지 큐 식별자 생성, 기존 SM 파일 열기, 새로운 SM 파일 생성

```
int i;
int smfd[SM_COUNT];
int sm_fd[SM_COUNT];
int msgq_id[SM_COUNT];

for (i = 0; i < SM_COUNT; ++i) {
    key_t key = MSG_KEY_BASE + i;
    if ((msgq_id[i] = msgget(key, IPC_CREAT | 0666)) == -1) {
        error_exit("msgget");
    }
}

char filename[20];
for (i = 0; i < SM_COUNT; ++i) {
    snprintf(filename, sizeof(filename), "SM%d", i + 1);
    if ((smfd[i] = open(filename, O_RDONLY)) == -1) {
        error_exit("open");
    }
}

for (i = 0; i < SM_COUNT; ++i) {
    snprintf(filename, sizeof(filename), "SM_%d", i + 1);
    if ((sm_fd[i] = open(filename, O_CREAT | O_WRONLY | O_TRUNC, 0644)) == -1) {
        error_exit("open");
    }
}
```

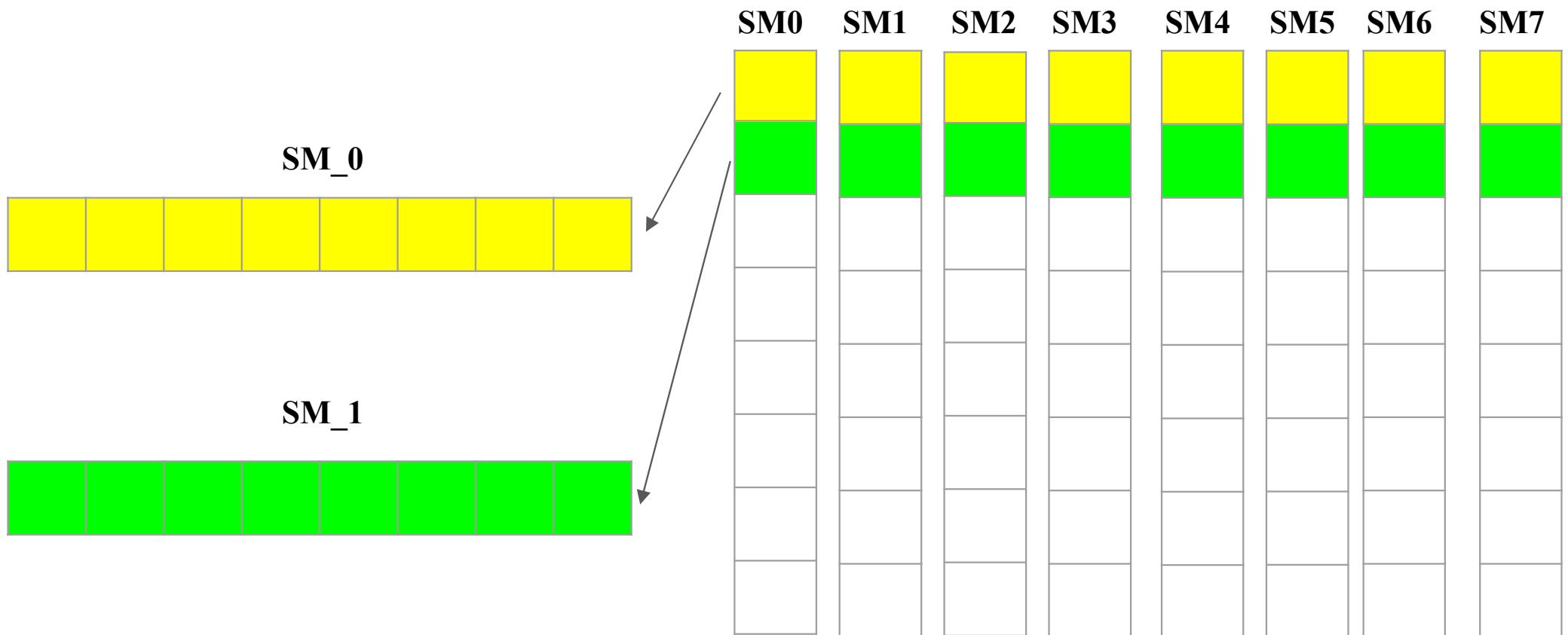
Key : 1000 ~ 1007

기존 SM 파일명 : SM0

새로운 SM 파일명 : SM_0

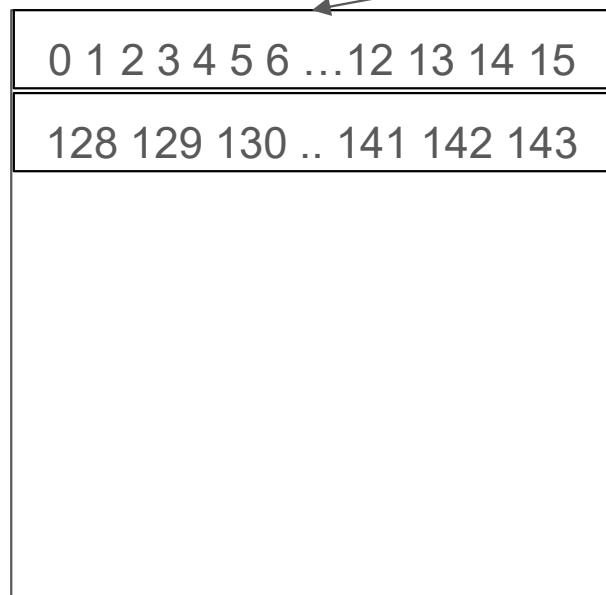
client - client , 8x8 Partitioning

main 함수



client - client , 8x8 Partitioning

main 함수

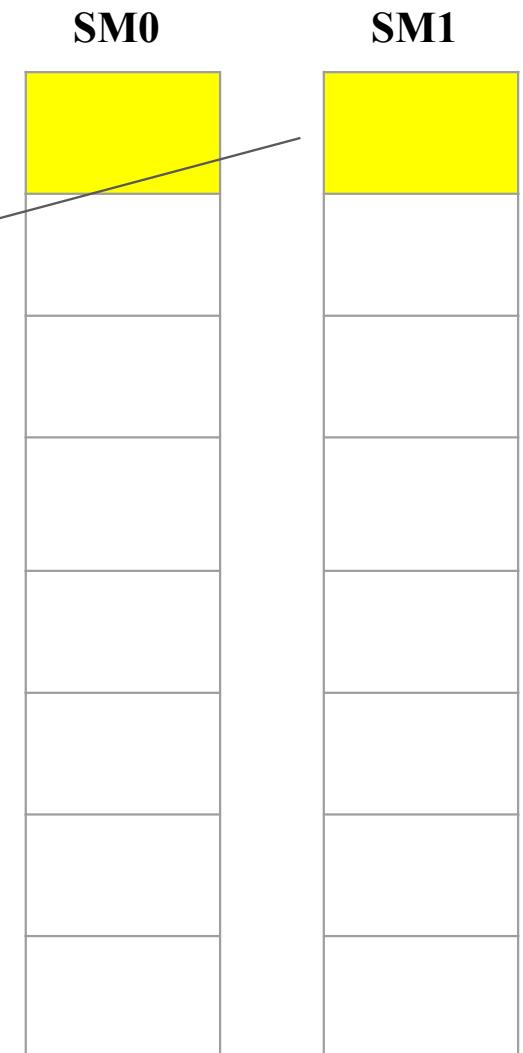


0
1
2

13
14
15

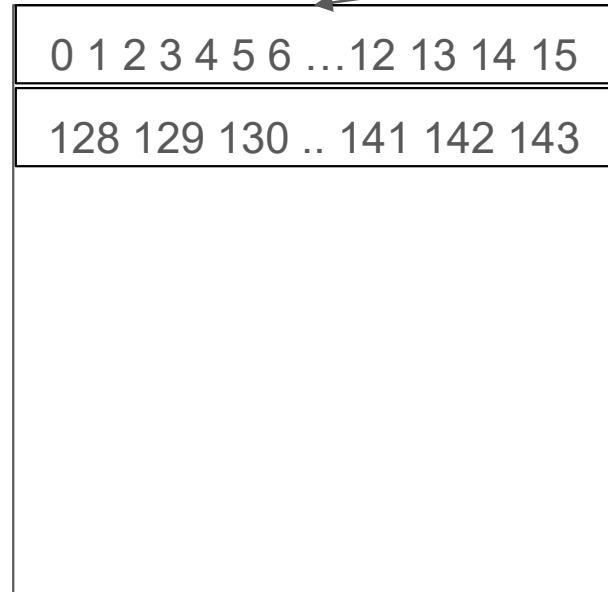


13
14
15



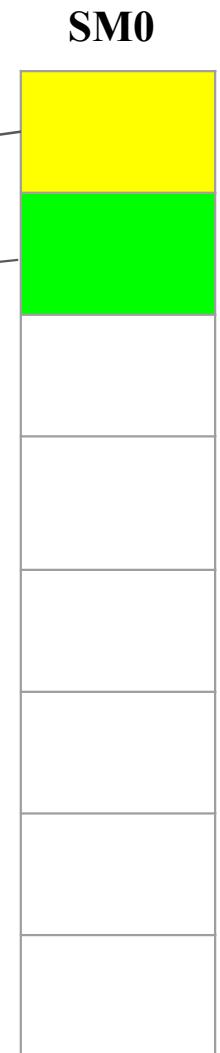
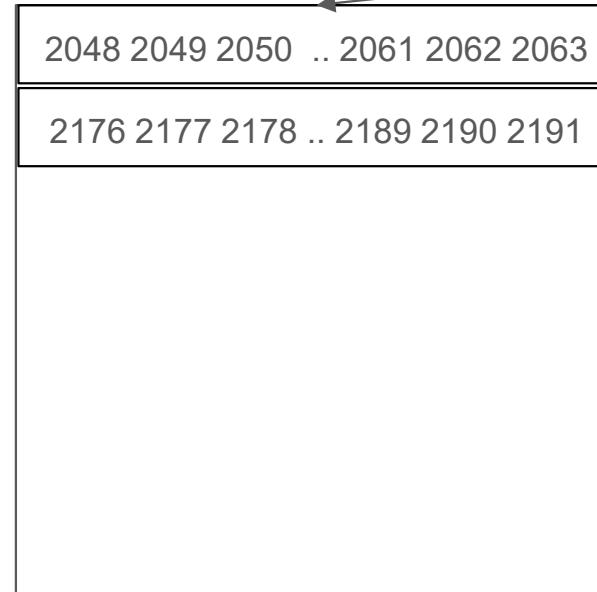
client - client , 8x8 Partitioning

main 함수



0
1
2

13
14
15



SM0

SM0

0 ~ 15	0번째 줄
2048 ~ 2063	16번째 줄
4096 ~ 4111	32번째 줄
6144 ~ 6159	48번째 줄
8192 ~ 8207	64번째 줄
10240 ~ 10255	80번째 줄
12288 ~ 12303	96번째 줄
14336 ~ 14351	112번째 줄

SM_0	
SM_1	
SM_2	
SM_3	
SM_4	
SM_5	
SM_6	
SM_7	

msgq_id[0]



msgq_id[1]



msgq_id[2]



msgq_id[3]



msgq_id[4]



msgq_id[5]



msgq_id[6]



msgq_id[7]



client - client , 8x8 Partitioning

main 함수 - 반복문

```
int j, k;
for (i = 0; i < ROWS_PER_FILE; ++i) {
    for (j = 0; j < SM_COUNT; ++j) {
        for (k = 0; k < SM_COUNT; ++k) {
```

i : SM의 각 블록 중 몇번째 줄인가? (0 ~ 15)

j : 몇번째 SM을 사용하는가? (0 ~ 7)

k : 몇번째 new SM에 옮겨야 하는가? (0 ~ 7)

client - client , 8x8 Partitioning

main 함수 - 메세지 송신

```
struct message msg;
msg.msg_type = 1;

int offset = (k * BLOCK_SIZE * ROWS_PER_FILE * sizeof(int)) + (i * BLOCK_SIZE * sizeof(int));

if (lseek(smfid[j], offset, SEEK_SET) == -1)
{
    close_(smfd, sm_fd, msgq_id);
    error_exit("lseek");
}

if (read(smfid[j], msg.data, sizeof(msg.data)) != sizeof(msg.data))
{
    close_(smfd, sm_fd, msgq_id);
    error_exit("read");
}

int SM_index = k;
if (msgsnd(msgq_id[SM_index], &msg, sizeof(msg.data), 0) == -1)
{
    close_(smfd, sm_fd, msgq_id);
    error_exit("msgsnd");
}
```

Offset ex)

**SM0 → SM_1 첫번째 16개 정수
(2048 ~ 2063)**
k = 1 , i = 0

offset

$$\begin{aligned} &= (1 * 16 * 16 * \text{sizeof}(int)) \\ &\quad + (0 * 16 * \text{sizeof}(int)) \\ &= 256 * \text{sizeof}(int) \\ &\rightarrow \text{두번째 블럭의 시작} \end{aligned}$$

client - client , 8x8 Partitioning

main 함수 - 메세지 수신

```
int SM_index = k;
if (msgsnd(msgq_id[SM_index], &msg, sizeof(msg.data), 0) == -1)
{
    close_(smfd, sm_fd, msgq_id);
    error_exit("msgsnd");
}

struct message received_msg;
if (msgrcv(msgq_id[SM_index], &received_msg, sizeof(received_msg.data), 1, 0) == -1)
{
    close_(smfd, sm_fd, msgq_id);
    error_exit("msgrcv");
}
if (write(sm_fd[SM_index], received_msg.data, sizeof(received_msg.data)) != sizeof(received_msg.data))
{
    close_(smfd, sm_fd, msgq_id);
    error_exit("write");
}
```

client - client , 8x8 Partitioning

프로그램 실행

```
[s20011914@sce 1204]$ ls  
8by8      SM1    SM3    SM5    SM7    makeSecondSM_3      matrix_file  
8by8.c    SM2    SM4    SM6    SM8    makeSecondSM_3.c    removeNewSM
```

```
[s20011914@sce 1204]$ ./makeSecondSM_3  
process execute time : 0.039093 sec  
[s20011914@sce 1204]$ ls  
8by8      SM2    SM5    SM8    SM_3   SM_6    makeSecondSM_3  
8by8.c    SM3    SM6    SM_1   SM_4   SM_7    makeSecondSM_3.c  
SM1      SM4    SM7    SM_2   SM_5   SM_8    matrix_file
```

프로세스 실행시간 : 0.039093 sec

Client-Client Communication

4 by 4

client - client , 4x4 Partitioning

```
[s20003212@sce practice]$ od -i SM_1
0000000 0 1 2 3
0000020 4 5 6 7
0000040 8 9 10 11
0000060 12 13 14 15
0000100 16 17 18 19
0000120 20 21 22 23
0000140 24 25 26 27
0000160 28 29 30 31
0000200 32 33 34 35
0000220 36 37 38 39
0000240 40 41 42 43
0000260 44 45 46 47
0000300 48 49 50 51
0000320 52 53 54 55
0000340 56 57 58 59
0000360 60 61 62 63
0000400 64 65 66 67
0000420 68 69 70 71
0000440 72 73 74 75
0000460 76 77 78 79
```

SM_0 Front

```
0017320 1972 1973 1974 1975
0017340 1976 1977 1978 1979
0017360 1980 1981 1982 1983
0017400 1984 1985 1986 1987
0017420 1988 1989 1990 1991
0017440 1992 1993 1994 1995
0017460 1996 1997 1998 1999
0017500 2000 2001 2002 2003
0017520 2004 2005 2006 2007
0017540 2008 2009 2010 2011
0017560 2012 2013 2014 2015
0017600 2016 2017 2018 2019
0017620 2020 2021 2022 2023
0017640 2024 2025 2026 2027
0017660 2028 2029 2030 2031
0017700 2032 2033 2034 2035
0017720 2036 2037 2038 2039
0017740 2040 2041 2042 2043
0017760 2044 2045 2046 2047
0020000
[s20003212@sce practice]$ █
```

SM_0 back

client - client , 4x4 Partitioning

```
[s20003212@sce practice]$ od -i SM_2  
0000000 2048 2049 2050 2051  
0000020 2052 2053 2054 2055  
0000040 2056 2057 2058 2059  
0000060 2060 2061 2062 2063  
0000100 2064 2065 2066 2067  
0000120 2068 2069 2070 2071  
0000140 2072 2073 2074 2075  
0000160 2076 2077 2078 2079  
0000200 2080 2081 2082 2083  
0000220 2084 2085 2086 2087  
0000240 2088 2089 2090 2091  
0000260 2092 2093 2094 2095  
0000300 2096 2097 2098 2099  
0000320 2100 2101 2102 2103  
0000340 2104 2105 2106 2107  
0000360 2108 2109 2110 2111  
0000400 2112 2113 2114 2115  
0000420 2116 2117 2118 2119  
0000440 2120 2121 2122 2123  
0000460 2124 2125 2126 2127
```

SM_1 Front

```
0017320 4020 4021 4022 4023  
0017340 4024 4025 4026 4027  
0017360 4028 4029 4030 4031  
0017400 4032 4033 4034 4035  
0017420 4036 4037 4038 4039  
0017440 4040 4041 4042 4043  
0017460 4044 4045 4046 4047  
0017500 4048 4049 4050 4051  
0017520 4052 4053 4054 4055  
0017540 4056 4057 4058 4059  
0017560 4060 4061 4062 4063  
0017600 4064 4065 4066 4067  
0017620 4068 4069 4070 4071  
0017640 4072 4073 4074 4075  
0017660 4076 4077 4078 4079  
0017700 4080 4081 4082 4083  
0017720 4084 4085 4086 4087  
0017740 4088 4089 4090 4091  
0017760 4092 4093 4094 4095  
0020000  
[s20003212@sce practice]$ █
```

SM_1 back

client - client , 4x4 Partitioning

[s20003212@sce practice]\$ od -i SM_3	0000000	4096	4097	4098	4099	0017320	6068	6069	6070	6071
	0000020	4100	4101	4102	4103	0017340	6072	6073	6074	6075
	0000040	4104	4105	4106	4107	0017360	6076	6077	6078	6079
	0000060	4108	4109	4110	4111	0017400	6080	6081	6082	6083
	0000100	4112	4113	4114	4115	0017420	6084	6085	6086	6087
	0000120	4116	4117	4118	4119	0017440	6088	6089	6090	6091
	0000140	4120	4121	4122	4123	0017460	6092	6093	6094	6095
	0000160	4124	4125	4126	4127	0017500	6096	6097	6098	6099
	0000200	4128	4129	4130	4131	0017520	6100	6101	6102	6103
	0000220	4132	4133	4134	4135	0017540	6104	6105	6106	6107
	0000240	4136	4137	4138	4139	0017560	6108	6109	6110	6111
	0000260	4140	4141	4142	4143	0017600	6112	6113	6114	6115
	0000300	4144	4145	4146	4147	0017620	6116	6117	6118	6119
	0000320	4148	4149	4150	4151	0017640	6120	6121	6122	6123
	0000340	4152	4153	4154	4155	0017660	6124	6125	6126	6127
	0000360	4156	4157	4158	4159	0017700	6128	6129	6130	6131
	0000400	4160	4161	4162	4163	0017720	6132	6133	6134	6135
	0000420	4164	4165	4166	4167	0017740	6136	6137	6138	6139
	0000440	4168	4169	4170	4171	0017760	6140	6141	6142	6143
	0000460	4172	4173	4174	4175	0020000				

[s20003212@sce practice]\$ █

SM_2 Front

SM_2 back

client - client , 4x4 Partitioning

```
[s20003212@sce practice]$ od -i SM_4  
0000000 6144 6145 6146 6147  
0000020 6148 6149 6150 6151  
0000040 6152 6153 6154 6155  
0000060 6156 6157 6158 6159  
0000100 6160 6161 6162 6163  
0000120 6164 6165 6166 6167  
0000140 6168 6169 6170 6171  
0000160 6172 6173 6174 6175  
0000200 6176 6177 6178 6179  
0000220 6180 6181 6182 6183  
0000240 6184 6185 6186 6187  
0000260 6188 6189 6190 6191  
0000300 6192 6193 6194 6195  
0000320 6196 6197 6198 6199  
0000340 6200 6201 6202 6203  
0000360 6204 6205 6206 6207  
0000400 6208 6209 6210 6211  
0000420 6212 6213 6214 6215  
0000440 6216 6217 6218 6219  
0000460 6220 6221 6222 6223
```

SM_3 Front

```
0017320 8116 8117 8118 8119  
0017340 8120 8121 8122 8123  
0017360 8124 8125 8126 8127  
0017400 8128 8129 8130 8131  
0017420 8132 8133 8134 8135  
0017440 8136 8137 8138 8139  
0017460 8140 8141 8142 8143  
0017500 8144 8145 8146 8147  
0017520 8148 8149 8150 8151  
0017540 8152 8153 8154 8155  
0017560 8156 8157 8158 8159  
0017600 8160 8161 8162 8163  
0017620 8164 8165 8166 8167  
0017640 8168 8169 8170 8171  
0017660 8172 8173 8174 8175  
0017700 8176 8177 8178 8179  
0017720 8180 8181 8182 8183  
0017740 8184 8185 8186 8187  
0017760 8188 8189 8190 8191  
0020000  
[s20003212@sce practice]$ █
```

SM_3 back

client - client , 4x4 Partitioning

```
[s20003212@sce practice]$ od -i SM_5
0000000 8192 8193 8194 8195
0000020 8196 8197 8198 8199
0000040 8200 8201 8202 8203
0000060 8204 8205 8206 8207
0000100 8208 8209 8210 8211
0000120 8212 8213 8214 8215
0000140 8216 8217 8218 8219
0000160 8220 8221 8222 8223
0000200 8224 8225 8226 8227
0000220 8228 8229 8230 8231
0000240 8232 8233 8234 8235
0000260 8236 8237 8238 8239
0000300 8240 8241 8242 8243
0000320 8244 8245 8246 8247
0000340 8248 8249 8250 8251
0000360 8252 8253 8254 8255
0000400 8256 8257 8258 8259
0000420 8260 8261 8262 8263
0000440 8264 8265 8266 8267
0000460 8268 8269 8270 8271
0017320 10164 10165 10166 10167
0017340 10168 10169 10170 10171
0017360 10172 10173 10174 10175
0017400 10176 10177 10178 10179
0017420 10180 10181 10182 10183
0017440 10184 10185 10186 10187
0017460 10188 10189 10190 10191
0017500 10192 10193 10194 10195
0017520 10196 10197 10198 10199
0017540 10200 10201 10202 10203
0017560 10204 10205 10206 10207
0017600 10208 10209 10210 10211
0017620 10212 10213 10214 10215
0017640 10216 10217 10218 10219
0017660 10220 10221 10222 10223
0017700 10224 10225 10226 10227
0017720 10228 10229 10230 10231
0017740 10232 10233 10234 10235
0017760 10236 10237 10238 10239
0020000
[s20003212@sce practice]$
```

SM_4 Front

SM_4 back

client - client , 4x4 Partitioning

```
[s20003212@sce practice]$ od -i SM_6
0000000 10240 10241 10242 10243
0000020 10244 10245 10246 10247
0000040 10248 10249 10250 10251
0000060 10252 10253 10254 10255
0000100 10256 10257 10258 10259
0000120 10260 10261 10262 10263
0000140 10264 10265 10266 10267
0000160 10268 10269 10270 10271
0000200 10272 10273 10274 10275
0000220 10276 10277 10278 10279
0000240 10280 10281 10282 10283
0000260 10284 10285 10286 10287
0000300 10288 10289 10290 10291
0000320 10292 10293 10294 10295
0000340 10296 10297 10298 10299
0000360 10300 10301 10302 10303
0000400 10304 10305 10306 10307
0000420 10308 10309 10310 10311
0000440 10312 10313 10314 10315
0000460 10316 10317 10318 10319
```

SM_5 Front

```
0017320 12212 12213 12214 12215
0017340 12216 12217 12218 12219
0017360 12220 12221 12222 12223
0017400 12224 12225 12226 12227
0017420 12228 12229 12230 12231
0017440 12232 12233 12234 12235
0017460 12236 12237 12238 12239
0017500 12240 12241 12242 12243
0017520 12244 12245 12246 12247
0017540 12248 12249 12250 12251
0017560 12252 12253 12254 12255
0017600 12256 12257 12258 12259
0017620 12260 12261 12262 12263
0017640 12264 12265 12266 12267
0017660 12268 12269 12270 12271
0017700 12272 12273 12274 12275
0017720 12276 12277 12278 12279
0017740 12280 12281 12282 12283
0017760 12284 12285 12286 12287
0020000
[s20003212@sce practice]$ █
```

SM_5 back

client - client , 4x4 Partitioning

```
[s20003212@sce practice]$ od -i SM_7  
0000000 12288 12289 12290 12291  
0000020 12292 12293 12294 12295  
0000040 12296 12297 12298 12299  
0000060 12300 12301 12302 12303  
0000100 12304 12305 12306 12307  
0000120 12308 12309 12310 12311  
0000140 12312 12313 12314 12315  
0000160 12316 12317 12318 12319  
0000200 12320 12321 12322 12323  
0000220 12324 12325 12326 12327  
0000240 12328 12329 12330 12331  
0000260 12332 12333 12334 12335  
0000300 12336 12337 12338 12339  
0000320 12340 12341 12342 12343  
0000340 12344 12345 12346 12347  
0000360 12348 12349 12350 12351  
0000400 12352 12353 12354 12355  
0000420 12356 12357 12358 12359  
0000440 12360 12361 12362 12363  
0000460 12364 12365 12366 12367
```

SM_6 Front

```
0017320 14260 14261 14262 14263  
0017340 14264 14265 14266 14267  
0017360 14268 14269 14270 14271  
0017400 14272 14273 14274 14275  
0017420 14276 14277 14278 14279  
0017440 14280 14281 14282 14283  
0017460 14284 14285 14286 14287  
0017500 14288 14289 14290 14291  
0017520 14292 14293 14294 14295  
0017540 14296 14297 14298 14299  
0017560 14300 14301 14302 14303  
0017600 14304 14305 14306 14307  
0017620 14308 14309 14310 14311  
0017640 14312 14313 14314 14315  
0017660 14316 14317 14318 14319  
0017700 14320 14321 14322 14323  
0017720 14324 14325 14326 14327  
0017740 14328 14329 14330 14331  
0017760 14332 14333 14334 14335  
0020000  
[s20003212@sce practice]$ █
```

SM_6 back

client - client , 4x4 Partitioning

```
[s20003212@sce practice]$ od -i SM_8
0000000    14336    14337    14338    14339
0000020    14340    14341    14342    14343
0000040    14344    14345    14346    14347
0000060    14348    14349    14350    14351
0000100    14352    14353    14354    14355
0000120    14356    14357    14358    14359
0000140    14360    14361    14362    14363
0000160    14364    14365    14366    14367
0000200    14368    14369    14370    14371
0000220    14372    14373    14374    14375
0000240    14376    14377    14378    14379
0000260    14380    14381    14382    14383
0000300    14384    14385    14386    14387
0000320    14388    14389    14390    14391
0000340    14392    14393    14394    14395
0000360    14396    14397    14398    14399
0000400    14400    14401    14402    14403
0000420    14404    14405    14406    14407
0000440    14408    14409    14410    14411
0000460    14412    14413    14414    14415
```

SM_7 Front

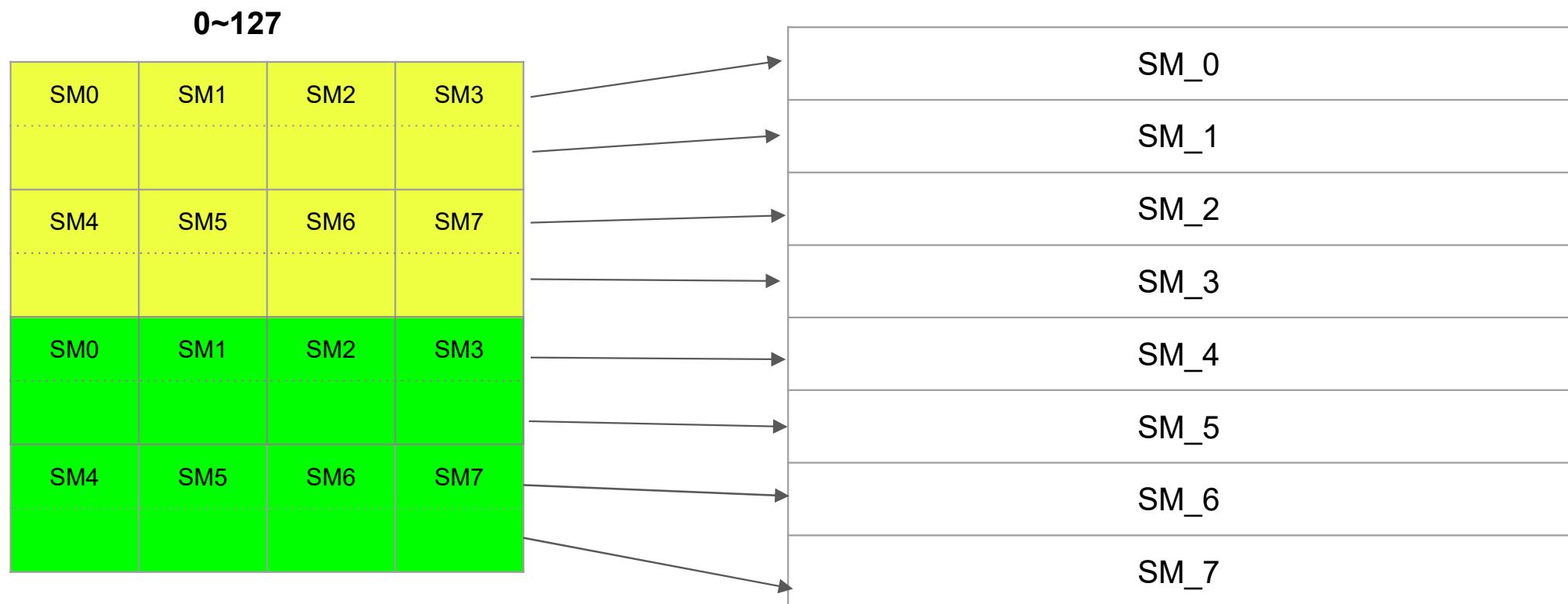
```
0017320    16308    16309    16310    16311
0017340    16312    16313    16314    16315
0017360    16316    16317    16318    16319
0017400    16320    16321    16322    16323
0017420    16324    16325    16326    16327
0017440    16328    16329    16330    16331
0017460    16332    16333    16334    16335
0017500    16336    16337    16338    16339
0017520    16340    16341    16342    16343
0017540    16344    16345    16346    16347
0017560    16348    16349    16350    16351
0017600    16352    16353    16354    16355
0017620    16356    16357    16358    16359
0017640    16360    16361    16362    16363
0017660    16364    16365    16366    16367
0017700    16368    16369    16370    16371
0017720    16372    16373    16374    16375
0017740    16376    16377    16378    16379
0017760    16380    16381    16382    16383
0020000
[s20003212@sce practice]$ █
```

SM_7 back

client - client , 4x4 Partitioning

main 함수

SM_0 ~ SM_7

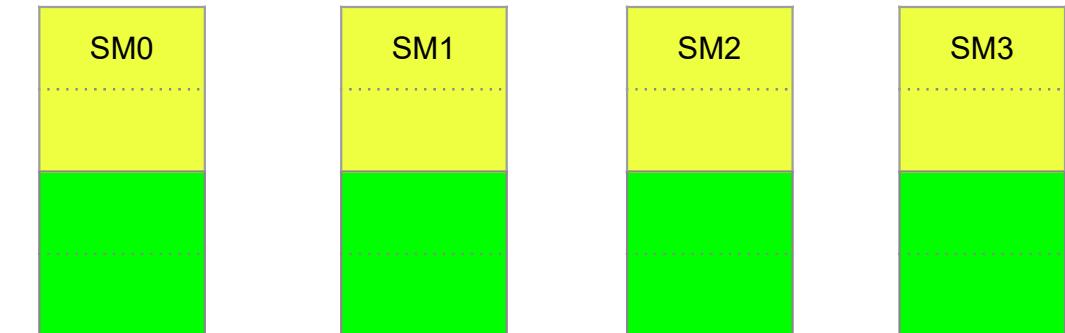


client - client , 4x4 Partitioning

main 함수

0~128

SM0	SM1	SM2	SM3
0~128			
SM4	SM5	SM6	SM7
0~128			
SM0	SM1	SM2	SM3
SM4	SM5	SM6	SM7

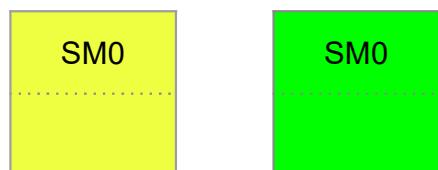


32개의 열

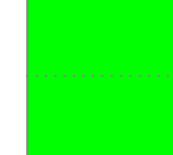
64개의 행

32개의 열

32개의 행

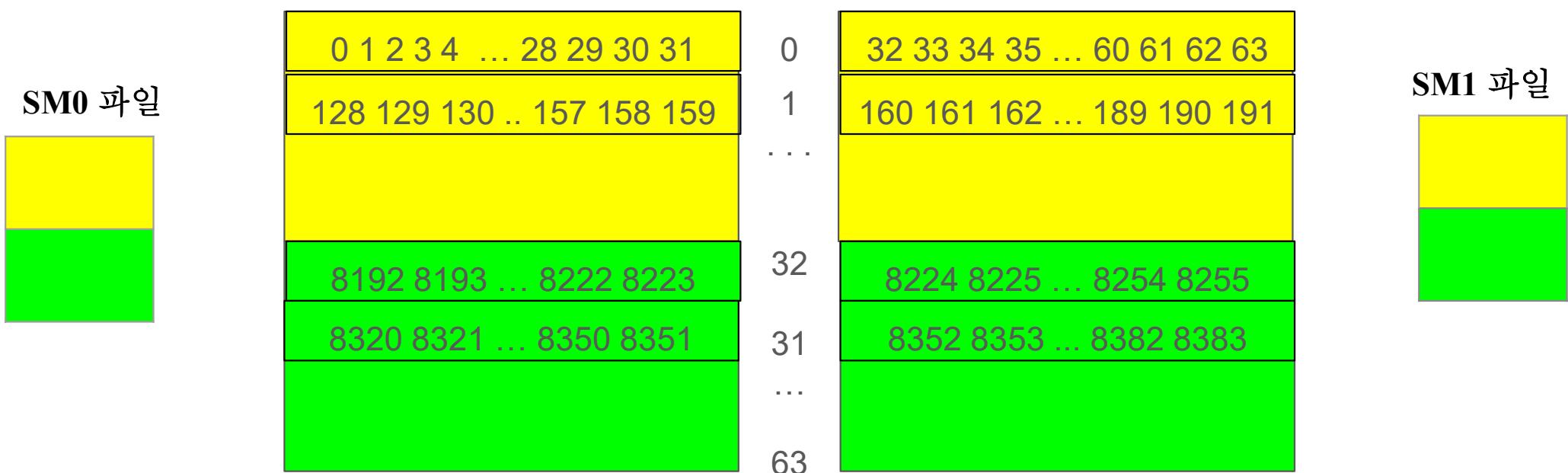


SM0



client - client , 4x4 Partitioning

main 함수



client - client , 4x4 Partitioning

```
#define BLOCK_SIZE 32
#define SM_COUNT 8
#define ROWS_PER_FILE 16
#define MSG_KEY_BASE 1000
```

BLOCK SIZE 32 : 128 X 128 을 16개 블럭으로 나눌 시 한
블럭 크기 32 X 32

SM_COUNT 8

ROWS_PER_FILE 16 : 1 Block = 16 rows

MSG_KEY_BASE 1000 : 키 생성 시 이용

client - client , 4x4 Partitioning

```
void error_exit(const char* msg) {
    perror(msg);
    exit(1);
}
```

오류 발생 메세지 출력 함수

```
//메시지 구조체
struct message {
    long msg_type;
    int data[BLOCK_SIZE];
};
```

메세지 구조체 : 정수형 배열 송/수신

```
void close_(int* smfd, int* sm_fd, int* msgq_id) {
    int i;
    for (i = 0; i < SM_COUNT; ++i) {
        close(smfd[i]);
        close(sm_fd[i]);
    }

    for (i = 0; i < SM_COUNT; ++i) {
        if (msgctl(msgq_id[i], IPC_RMID, NULL) == -1) {
            perror("메시지 큐 삭제 실패");
        }
    }
}
```

메세지 큐 & 기존 SM 파일, 새로운 SM 파일 닫기

client - client , 4x4 Partitioning

main 함수- 프로세스 시간 측정

```
struct timeval start, end;
long seconds, microseconds;
double elapsed;

gettimeofday(&start, NULL);
```

```
gettimeofday(&end, NULL);

seconds = end.tv_sec - start.tv_sec;
microseconds = end.tv_usec - start.tv_usec;

elapsed = seconds + microseconds * 1e-6;

printf("프로세스 실행 시간 : %.6f 초 \n", elapsed);
```

송/수신 프로세스 시작 전 시각 측정

송/수신 프로세스 후 시각 측정

종료 시각 - 시작 시각 = 프로세스 실행 시간

client - client , 4x4 Partitioning

main 함수 - 메세지 큐 식별자 생성, 기존 SM 파일 열기, 새로운 SM 파일 생성

Key : 1000 ~ 1007

```
// 1. 메시지 큐 생성
for (i = 0; i < SM_COUNT; ++i) {
    key_t key = MSG_KEY_BASE + i;
    if ((msgq_id[i] = msgget(key, IPC_CREAT | 0666)) == -1) {
        error_exit("메시지 큐 생성 실패");
    }
}
```

client - client , 4x4 Partitioning

```
//2. SM 파일을 읽기전용으로 열기
char filename[20];
for (i = 0; i < SM_COUNT; ++i) {
    snprintf(filename, sizeof(filename), "SM%d", i + 1);
    if ((smfd[i] = open(filename, O_RDONLY)) == -1) {
        error_exit("SM 파일 열기 실패");
    }
}
```

기존 SM 파일명 : SM0

```
//3. SM_ 파일 생성 및 쓰기전용으로 열기
for (i = 0; i < SM_COUNT; ++i) {
    snprintf(filename, sizeof(filename), "SM_%d", i + 1);
    if ((sm_fd[i] = open(filename, O_CREAT | O_WRONLY | O_TRUNC, 0644)) == -1) {
        error_exit("SM_ 파일 열기 실패");
    }
}
```

새로운 SM 파일명 : SM_0

client - client , 4x4 Partitioning

main 함수 - 반복문

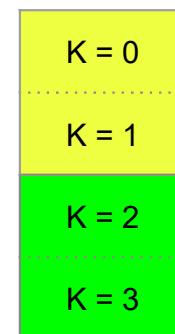
```
for (i = 0; i < ROWS_PER_FILE; ++i) {  
    for (j = 0; j < SM_COUNT; ++j) {  
        for (k = 0; k < 4; ++k) {
```

i : 각 SM 파트에서부터 몇번째 줄의 정수를 이동하는가? (0 ~ 15)

j : 몇번째 SM을 사용하는가? (0 ~ 7)

k : 해당하는 SM의 어느 파트인가? (0~3)

k=1인경우, 이 파트에서부터
i번째 행을 통신



j번째 SM파일

main 함수 - 메세지 송신

```
struct message msg;
msg.msg_type = 1;

int offset =
(k * BLOCK_SIZE * ROWS_PER_FILE * sizeof(int)) + (i * BLOCK_SIZE * sizeof(int));

if (lseek(smfid[j], offset, SEEK_SET) == -1)
{
    printf("sm : %d\nj : %d\nk : %d\n", j, i, k);
    close_(smfd, sm_fd, msgq_id);
    error_exit("lseek 실패");
}

if (read(smfid[j], msg.data, sizeof(msg.data)) != sizeof(msg.data))
{
    printf("sm : %d\nj : %d\nk : %d\n", j, i, k);
    close_(smfd, sm_fd, msgq_id);
    error_exit("SM 데이터 읽기 실패");
}

int SM_index;
if(k < 2) SM_index = ((j / 4) * 2) + k;
else SM_index = ((j / 4) * 2) + k + 2;

if (msgsnd(msgq_id[SM_index], &msg, sizeof(msg.data), 0) == -1)
{
    printf("sm : %d\nj : %d\nk : %d\n", j, i, k);
    close_(smfd, sm_fd, msgq_id);
    error_exit("메시지 큐 전송 실패");
}
```

1. offset은 i와 k로 지정
2. 32개의 정수를 msg 메세지 큐에 담는다.
3. 내용을 받을 SM_ 파일을 j 와 k 를 이용하여 변수 SM_index로 지정
4. 미리 만들어둔 msgq_id[]와 SM_index, msg를 사용하여 msgsnd 함수로 송신

client - client , 4x4 Partitioning

main 함수 - 메세지 수신

```
struct message received_msg;
if (msgrcv(msgq_id[SM_index], &received_msg, sizeof(received_msg.data), 1, 0) == -1)
{
    printf("sm : %d\n열 : %d\nk : %d\n", j, i, k);
    close_(smfd, sm_fd, msgq_id);
    error_exit("메시지 큐 수신 실패");
}
if (write(sm_fd[SM_index], received_msg.data, sizeof(received_msg.data)) != sizeof(received_msg.data))
{
    printf("sm : %d\n열 : %d\nk : %d\n", j, i, k);
    close_(smfd, sm_fd, msgq_id);
    error_exit("SM_데이터 쓰기 실패");
}
```

1. 수신할 메세지 큐 received_msg 생성
2. msgq_id[SM_index], received_msg를 인자로 하여 received_msg에 수신
3. 수신한 내용을 SM_파일에 저장

client - client , 4x4 Partitioning

프로그램 실행

```
[s20003212@sce practice]$ ls
SM1  SM2  SM3  SM4  SM5  SM6  SM7  SM8  dd  madeSecondSM  madeSecondSM.c
[s20003212@sce practice]$ ./madeSecondSM
프로세스 실행 시간 : 0.028540 초
[s20003212@sce practice]$ ls
SM1  SM3  SM5  SM7  SM_1  SM_3  SM_5  SM_7  dd
SM2  SM4  SM6  SM8  SM_2  SM_4  SM_6  SM_8  madeSecondSM
[s20003212@sce practice]$
```

프로세스 실행시간 : 0.028540 sec

Client-Server Communication

UNIX Team Project

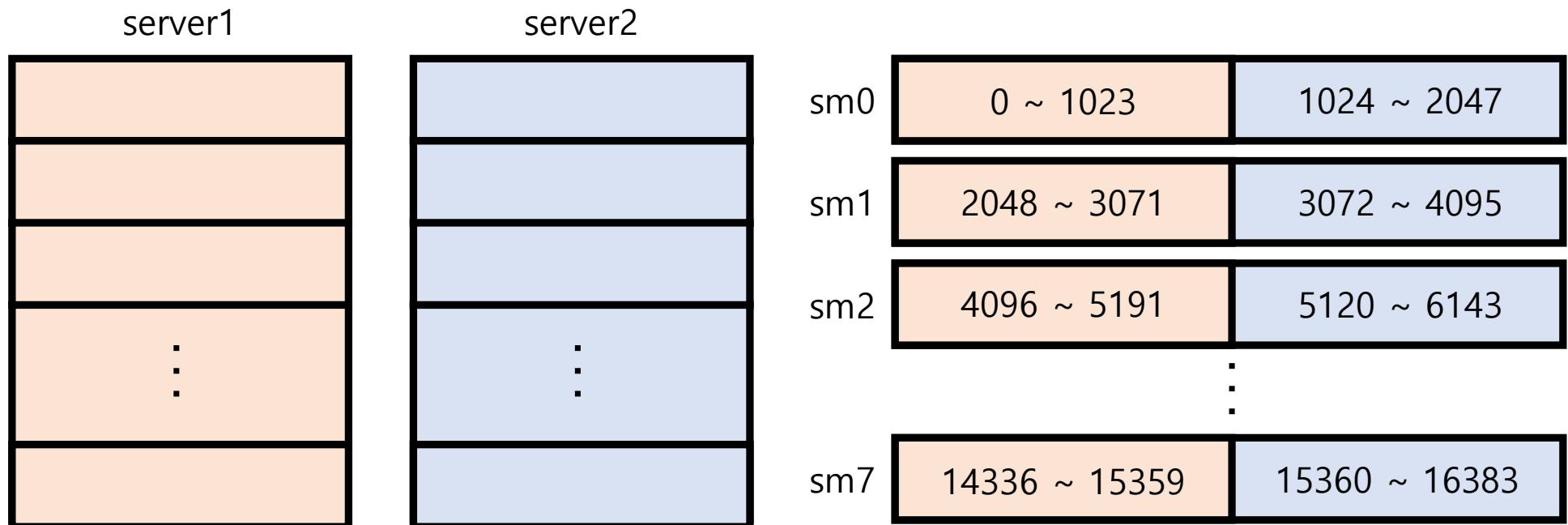
2024 – 2 UNIX Programming

Client – Server Communication

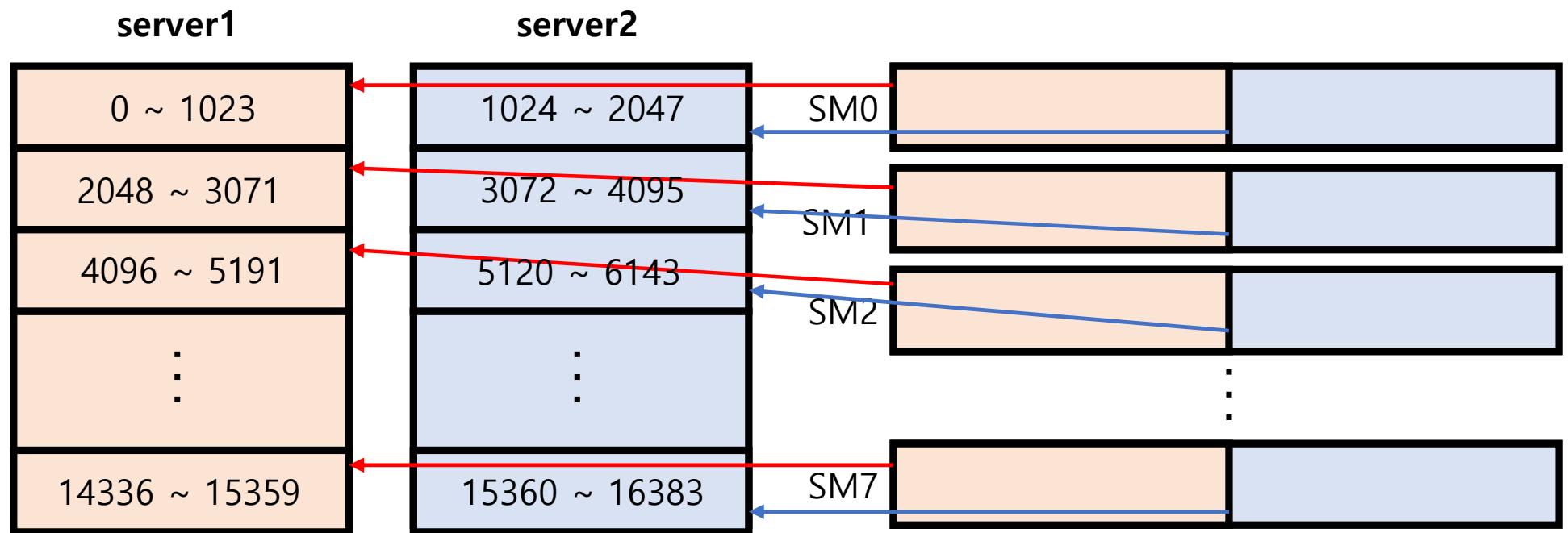
I/O block size : 4096 bytes(1024개의 정수)

Client – Client 통신을 마친 SM이 가진 정수 : 2048개 - 2개로 분할하여 통신

통신 기법 : `select()` 함수 사용을 위해 파이프 통신을 선택



Client – Server Communication



Client – Server Communication

```
int client(int sm, int p1[2], int p2[2]) {
    int n;
    int fd;
    char buffer[1024 * sizeof(int)];
    char filename[256];

    close(p1[0]); close(p2[0]); // 읽기 끝을 닫음 (쓰기만 수행)
    snprintf(filename, sizeof(filename), "client_file_%d", sm);
    fd = open(filename, O_RDONLY);
    if(fd == -1){
        printf("open error");
    }
    n = read(fd, buffer, 1024 * sizeof(int));
    printf("client %d / read bytes from file = %d\n", sm, n);
    write(p1[1],buffer, n);
    n = read(fd, buffer, 1024 * sizeof(int));
    printf("client %d / read bytes from file = %d\n", sm, n);
    write(p2[1],buffer, n);

    close(fd);
    exit(0);
}
```

p1[2] : server1과의 통신을 위한 fd
p2[2] : server2과의 통신을 위한 fd

첫 write는 p1을 사용,
두 번째 write는 p2를 사용해서 통신

Client – Server Communication

```
void server(int srvNum, int p1[8][2], int p2[8][2]) {
    char buf[8][MSGSIZE], ch; // 메시지를 저장할 버퍼
    fd_set set, master1, master2; // 파일 디스크립터 집합
    int i, fd, n, all_read;
    int readFlags[8] = {0};
    struct timeval tv1, tv2, tv3;

    // 모든 파이프의 쓰기 끝을 닫음 (읽기만 수행)
    for (i = 0; i < 8; i++) {
        close(p1[i][1]);
        close(p2[i][1]);
    }
}
```

■ p1[8][2]

server1이 모든 SM들과 통신할 때
사용하는 fd들로 이루어진 배열

■ p2[8][2]

server2가 모든 sm들과 통신할 때
사용하는 fd들로 이루어진 배열

MSGSIZE = 4096 (1024 integer)

Client – Server Communication

```
FD_ZERO(&master1); FD_ZERO(&master2); // master 집합 초기화

for (i = 0; i < 8; i++) FD_SET(p1[i][0], &master1);

// select를 사용하여 파일 디스크립터 상태를 확인
while (set = master1, select(p1[7][0] + 1, &set, NULL, NULL, NULL) > 0) {

    gettimeofday(&tv1, NULL);

    for (i = 0; i < 8; i++) {
        if (FD_ISSET(p1[i][0], &set)) { // 특정 파일에서 읽을 준비가 됐는지 확인
            if ((n = read(p1[i][0], buf[i], MSGSIZE)) > 0) { // 메시지 읽기
                printf("server1 / data receiving... i = %d, read %d bytes\n", i, n);
                readFlags[i] = 1;
            }
        }
    }

    all_read = 1;
    for (i = 0; i < 8; i++) {
        if (readFlags[i] == 0) {
            all_read = 0;
            break;
        }
    }
}
```

서버는 **select()** 함수를 이용해서
모든 SM들로부터 데이터를 받아들일 준비

server1은 p1에서만 데이터 통신

server2는 p2에서만 데이터 통신

메세지를 읽으면 buf에 임시 저장
무작위 순서로 read되기 때문에
buf에 저장 후
알맞은 순서로 최종 파일에 write

Client – Server Communication

```
if (all_read){
    gettimeofday(&tv2, NULL);
    fd = open("final_result_1", O_RDWR|O_CREAT|O_TRUNC, 0644);
    for(i = 0; i < 8; i++){
        n = write(fd, buf[i], sizeof(buf[i]));
        printf("server1 / all data received. i = %d, write %d bytes\n", i, n);
    }
    close(fd);
```

```
long sec1 = tv2.tv_sec - tv1.tv_sec;
long sec2 = tv3.tv_sec - tv2.tv_sec;
long micro1 = tv2.tv_usec - tv1.tv_usec;
long micro2 = tv3.tv_usec - tv2.tv_usec;

double elapsed1 = sec1 + micro1 * 1e-6;
printf("server1 - client 통신 시간 : %.6f 초\n", elapsed1);
double elapsed2 = sec2 + micro2 * 1e-6;
printf("server1 IO 시간 : %.6f 초\n", elapsed2);
```

**모든 SM과의 통신을 마치면
파일에 write**

※ 실행시간 측정

server - client 통신 : 약 0.000015 초

server IO : 약 0.000150 초

Client – Server Communication

```
(base) geonwoo 🐻 ~/Desktop/unix_project ➤ od -i final_result_1 | more
0000000      0      1      2      3
0000020      4      5      6      7
0000040      8      9     10     11
0000060     12     13     14     15
0000100     16     17     18     19
0000120     20     21     22     23
0000140     24     25     26     27
```

```
0007660    1004    1005    1006    1007
0007700    1008    1009    1010    1011
0007720    1012    1013    1014    1015
0007740    1016    1017    1018    1019
0007760    1020    1021    1022    1023
0010000    2048    2049    2050    2051
0010020    2052    2053    2054    2055
0010040    2056    2057    2058    2059
0010060    2060    2061    2062    2063
0010100    2064    2065    2066    2067
0010120    2068    2069    2070    2071
```

```
0077620    15332   15333   15334   15335
0077640    15336   15337   15338   15339
0077660    15340   15341   15342   15343
0077700    15344   15345   15346   15347
0077720    15348   15349   15350   15351
0077740    15352   15353   15354   15355
0077760    15356   15357   15358   15359
```

```
(base) geonwoo 🐻 ~/Desktop/unix_project ➤ od -i final_result_2 | more
0000000      1024    1025    1026    1027
0000020      1028    1029    1030    1031
0000040      1032    1033    1034    1035
0000060      1036    1037    1038    1039
0000100      1040    1041    1042    1043
0000120      1044    1045    1046    1047
```

0007640	2024	2025	2026	2027
0007660	2028	2029	2030	2031
0007700	2032	2033	2034	2035
0007720	2036	2037	2038	2039
0007740	2042	2041	2042	2042
0007760	2044	2045	2046	2047
0010000	3072	3073	3074	3075
0010020	3076	3077	3078	3079
0010040	3080	3081	3082	3083
0010060	3084	3085	3086	3087

0017700	4080	4081	4082	4083
0017720	4084	4085	4086	4087
0017740	4088	4089	4090	4091
0017760	4092	4093	4094	4095
0020000	5120	5121	5122	5123
0020020	5124	5125	5126	5127
0020040	5128	5129	5130	5131

0077640	16360	16361	16362	16363
0077660	16364	16365	16366	16367
0077700	16368	16369	16370	16371
0077720	16372	16373	16374	16375
0077740	16376	16377	16378	16379
0077760	16380	16381	16382	16383