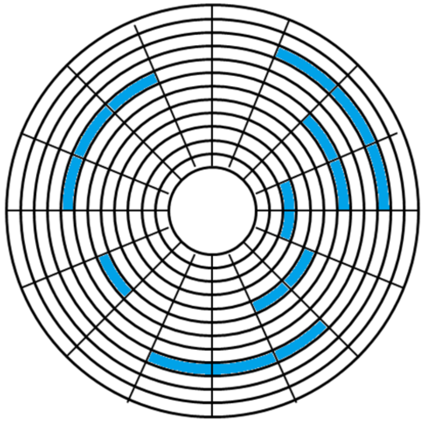
Operating Systems
Programming Assignment #7

File Fragmentation Reproduction

Prof. Li-Pin Chang, NCTU

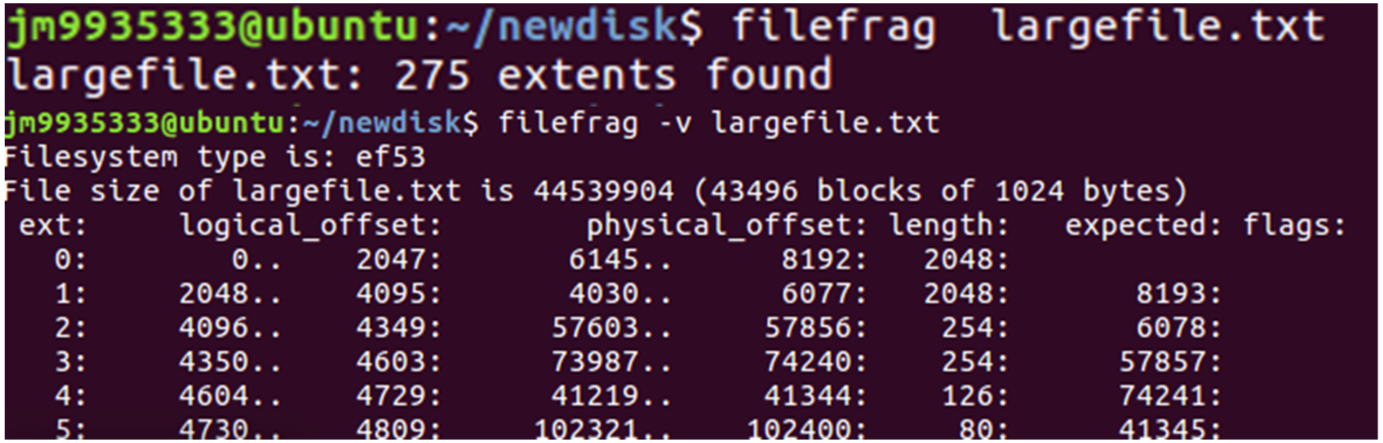
File fragmentation

- Ideally, a file is allocated in contiguous disk space
  - File system ages after it undergoes a number of allocation and de-allocation requests
  - Free space become fragmented
  - It is difficult for an aged file system to allocate contiguous disk space for a new file or an existing to grow
- File fragmentation increase the disk positioning overhead



Extents

- An extent = multiple contiguous disk blocks
  - Ext4 allocates in terms of extents to avoid file fragmentation
  - With delayed allocation and pre-allocation
- You can use <filefrag -v filename> command to get the information of the extents of a file



```
jm9935333@ubuntu:~/newdisk$ filefrag  largefile.txt
largefile.txt: 275 extents found
jm9935333@ubuntu:~/newdisk$ filefrag -v largefile.txt
Filesystem type is: ef53
File size of largefile.txt is 44539904 (43496 blocks of 1024 bytes)
 ext:     logical_offset:        physical_offset: length:   expected: flags:
   0:        0..    2047:       6145..      8192:   2048:
   1:     2048..    4095:       4030..      6077:   2048:       8193:
   2:     4096..    4349:      57603..     57856:    254:       6078:
   3:     4350..    4603:      73987..     74240:    254:      57857:
   4:     4604..    4729:      41219..     41344:    126:      74241:
   5:     4730..    4809:     102321..    102400:     80:      41345:
```

APIs & Tools

- fopen(), fclose(), fread(), fwrite(), fsync(), etc

- filefrag [-Bbsvx] [files……]
  - Reports the details of a file’s extents
  - https://linux.die.net/man/8/filefrag

Procedure

- Your program
  - gets time t1
  - runs on a 100 MB disk volume formatted in ext4
  - ages the 100 MB file system with your own method
  - produces a file “largefile.txt”
  - system(“filefrag -v largefile.txt”);
  - gets time t2
  - prints t2-t1 (in seconds)

Requirements

- “largefile.txt” must have >= 400 extents
- The virtual disk contains only your program and files created by it
  - The sizes of created new files must be powers of 2
- Your program must not run too slow, see the grading policy below

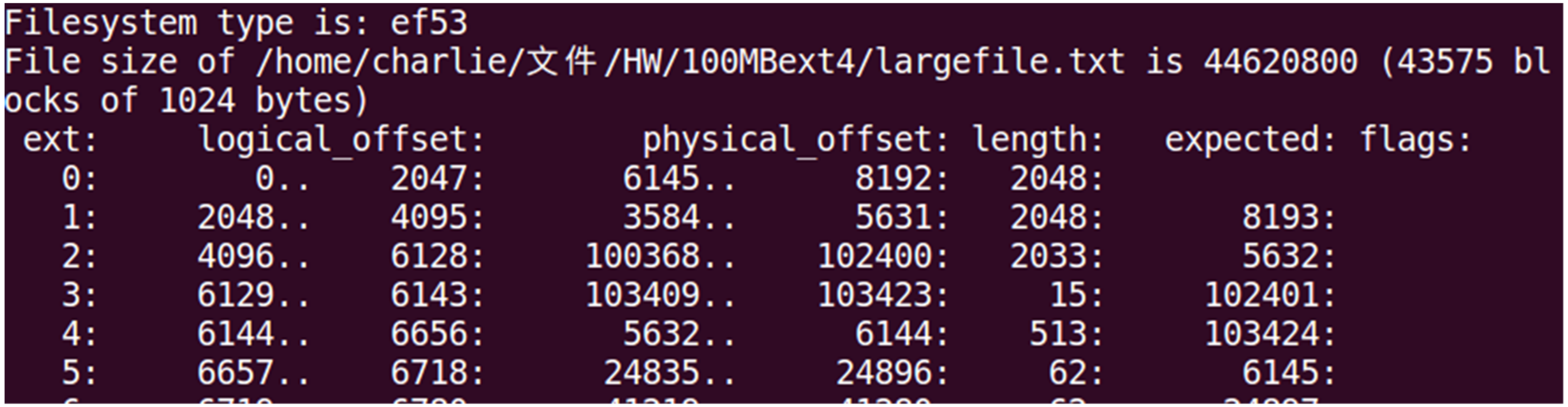
Test Flow

1. Attach a 100 MB virtual disk to your VM
2. Format the disk in ext4
3. Mount the disk
4. Copy your program to the new volume
5. Run your program
    - Age the file system with your own method…
    - Create “largefile.txt”
    - Call system(“filefrag -v largefile.txt”)
    - Print elapsed time in xxx.xx seconds

Grading

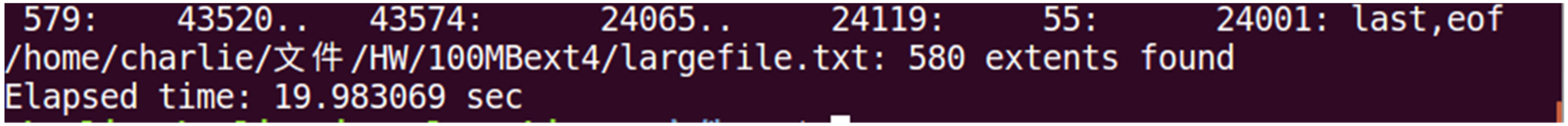
- 30% meet all the requirements
  - Program flow, file name “largfile.txt”, file sizes (powers of 2)…
- 30% extent count of “largefile.txt”
  - 30 pts if >= 400
  - 20 pts if in [200, 400]
  - 0 pts if < 200
- 40% program efficiency
  - Let X=the median execution time of all students’ programs
  - Let Y=the execution of your program, you will get
  - 40 pts if Y <=X
  - 35 pts if Y is in (X,2X]
  - 20 pts if Y >2X
- Extent # and execution times will be measured on the TA’s platform

Example



```
Filesystem type is: ef53
File size of /home/charlie/文件/HW/100MBext4/largefile.txt is 44620800 (43575 bl
ocks of 1024 bytes)
 ext:     logical_offset:        physical_offset: length:   expected: flags:
   0:        0..    2047:       6145..      8192:   2048:
   1:     2048..    4095:       3584..      5631:   2048:       8193:
   2:     4096..    6128:     100368..    102400:   2033:       5632:
   3:     6129..    6143:     103409..    103423:     15:     102401:
   4:     6144..    6656:       5632..      6144:    513:     103424:
   5:     6657..    6718:      24835..     24896:     62:       6145:
```

⋮



```
 579:    43520..   43574:      24065..     24119:     55:      24001: last,eof
/home/charlie/文件/HW/100MBext4/largefile.txt: 580 extents found
Elapsed time: 19.983069 sec
```

Testing OS Environment

- Ubuntu 16.04, Ubuntu 14.04 or CS linux work station
    - Your code should compile successfully in one of the above environments