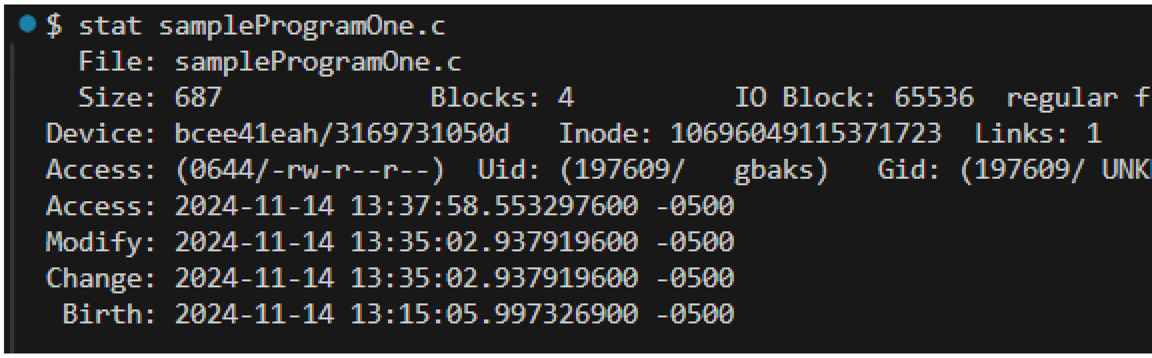
**CIS 452 02 – Lab 11 – Gabrielle Munson and Gabe Baksa**

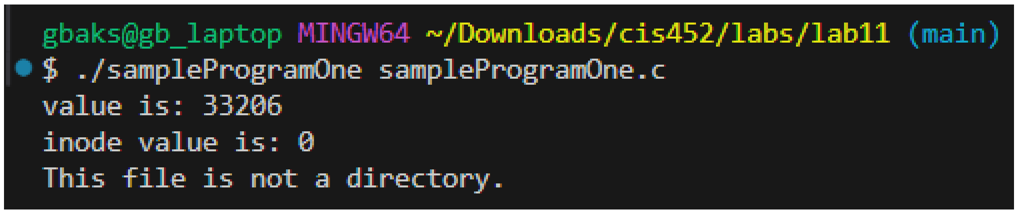
## Sample Program One

1. **Perform the following operations and answer the associated questions:**
   a) **Access the man pages, what is the difference between stat(1) and stat(2)?**
   stat(1) is a command-line tool for users to access file metadata whereas stat(2) is a system call for programs to retrieve and use file metadata within code.
   b) **Compile and test sampleProgramOne. Run it twice: use the sampleProgramOne source code file and then its executable file as test inputs. What exactly does sampleProgramOne do?**
   The first value output from the program is statBuf.st_mode which represents the file mode (including both the file type and permissions) as an unsigned integer. The second value the program outputs is statBuf.st_ino which is the inode number, a unique identifier for the file on the filesystem.
   c) **Modify sampleProgramOne so that it reports whether a file is a directory or not.**
      - **Verify that your program works (as shown below). Include a screenshot of the execution. Also include your source code as an attachment when uploading to blackboard.**
      - **Use sampleProgramOne and your current directory as your test inputs. Verify and demonstrate the correctness of your program by testing its output against the output of the stat(1) utility. For example:**
        - **stat sampleProgramOne.c**



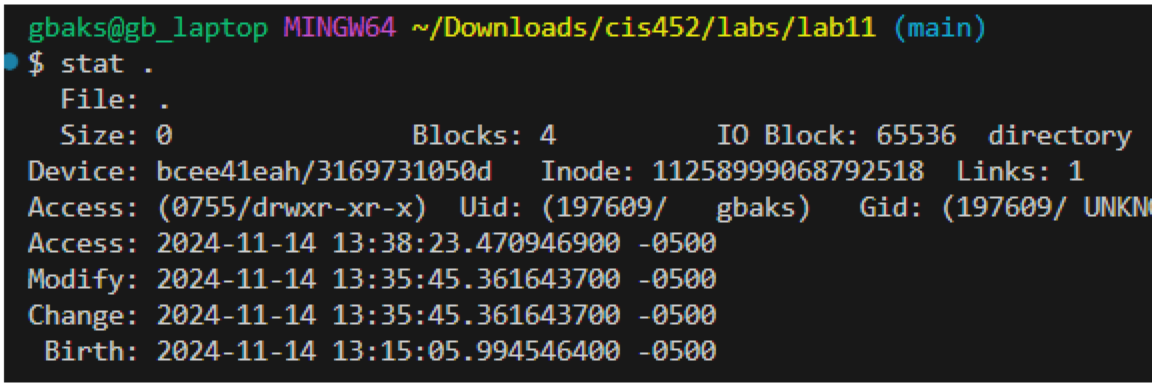
```
$ stat sampleProgramOne.c
  File: sampleProgramOne.c
  Size: 687             Blocks: 4          IO Block: 65536   regular f
Device: bcee41eah/3169731050d   Inode: 10696049115371723  Links: 1
Access: (0644/-rw-r--r--)  Uid: (197609/   gbaks)   Gid: (197609/ UNK
Access: 2024-11-14 13:37:58.553297600 -0500
Modify: 2024-11-14 13:35:02.937919600 -0500
Change: 2024-11-14 13:35:02.937919600 -0500
 Birth: 2024-11-14 13:15:05.997326900 -0500
```

        - **./sampleProgramOne sampleProgramOne.c**



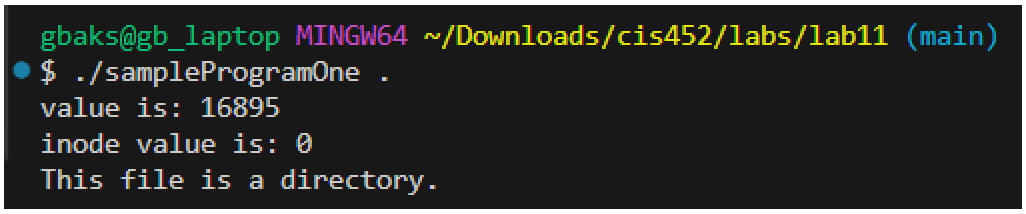
```
gbaks@gb_laptop MINGW64 ~/Downloads/cis452/labs/lab11 (main)
$ ./sampleProgramOne sampleProgramOne.c
value is: 33206
inode value is: 0
This file is not a directory.
```

        - **stat .**



```
gbaks@gb_laptop MINGW64 ~/Downloads/cis452/labs/lab11 (main)
$ stat .
  File: .
  Size: 0               Blocks: 4          IO Block: 65536  directory
Device: bcee41eah/3169731050d   Inode: 1125899906879251 8  Links: 1
Access: (0755/drwxr-xr-x)  Uid: (197609/   gbaks)   Gid: (197609/ UNKN
Access: 2024-11-14 13:38:23.470946900 -0500
Modify: 2024-11-14 13:35:45.361643700 -0500
Change: 2024-11-14 13:35:45.361643700 -0500
 Birth: 2024-11-14 13:15:05.994546400 -0500
```

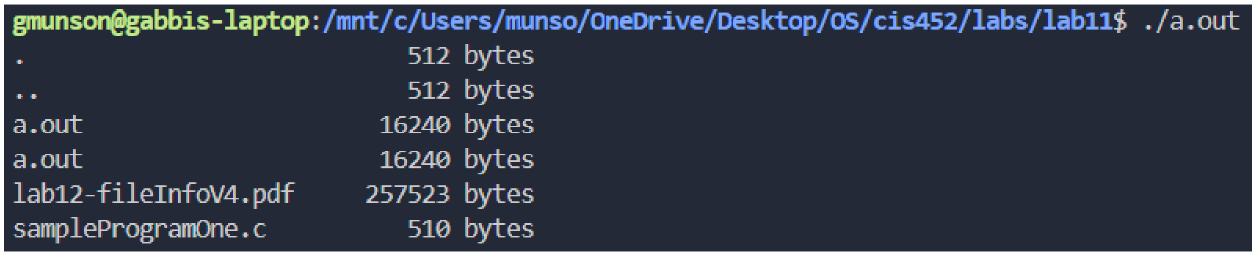
- **./sampleProgramOne .**



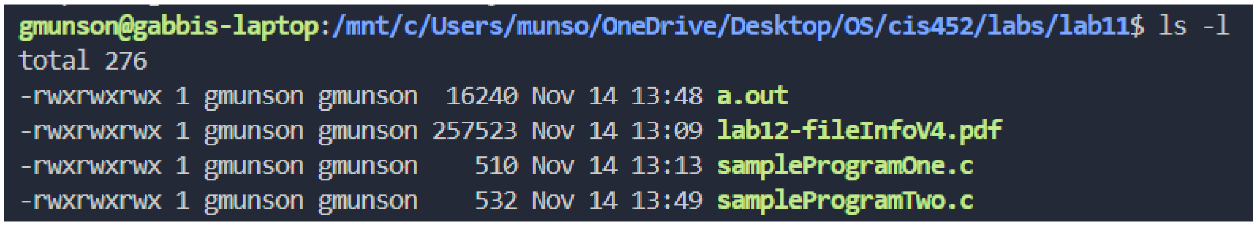
```
gbaks@gb_laptop MINGW64 ~/Downloads/cis452/labs/lab11 (main)
$ ./sampleProgramOne .
value is: 16895
inode value is: 0
This file is a directory.
```

## Sample Program 2

2. **Perform the following operations and answer the associated questions:**
   a) **Compile and test sampleProgramTwo, what exactly does sampleProgramTwo do?**
   SampleProgramTwo lists all entries in the current directory, specified/output as ".", and includes the parent directory which is output as "..". Each entry is printed on a new line
   b) **Modify sampleProgramTwo so that it also reports the size of each file (in bytes). Ensure your output is human-friendly (i.e., readable)**
   c) **Verify and demonstrate the correctness of your program by testing it against the *ls* program using your current directory. For example '*ls -l*' should return the same value as './sampleProgamTwo'. Submit your modified program and include a screenshot of the execution.**



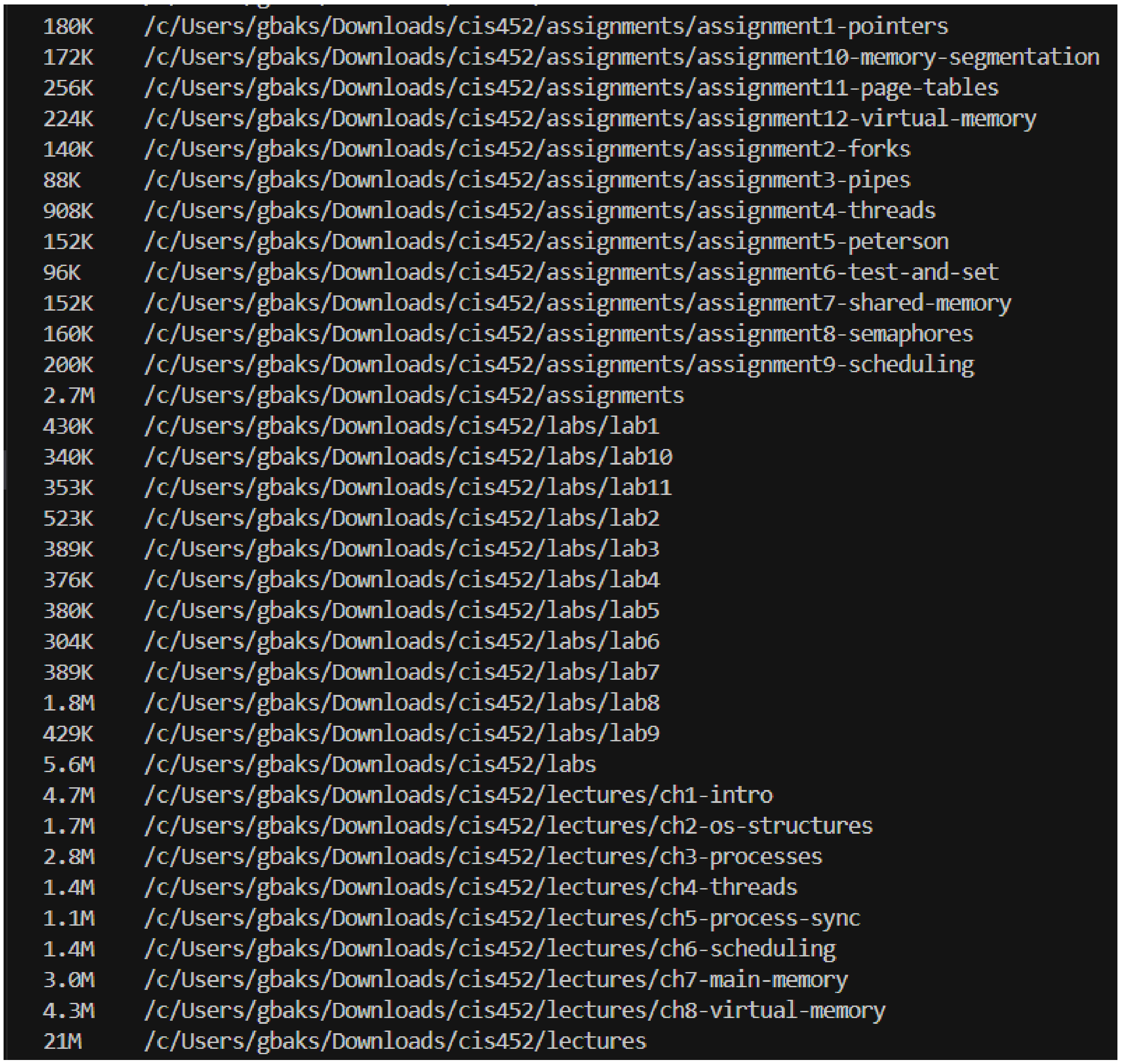
```
gmunson@gabbis-laptop:/mnt/c/Users/munso/OneDrive/Desktop/OS/cis452/labs/lab11$ ./a.out
.                          512 bytes
..                         512 bytes
a.out                    16240 bytes
a.out                    16240 bytes
lab12-fileInfoV4.pdf    257523 bytes
sampleProgramOne.c         510 bytes
```



```
gmunson@gabbis-laptop:/mnt/c/Users/munso/OneDrive/Desktop/OS/cis452/labs/lab11$ ls -l
total 276
-rwxrwxrwx 1 gmunson gmunson  16240 Nov 14 13:48 a.out
-rwxrwxrwx 1 gmunson gmunson 257523 Nov 14 13:09 lab12-fileInfoV4.pdf
-rwxrwxrwx 1 gmunson gmunson    510 Nov 14 13:13 sampleProgramOne.c
-rwxrwxrwx 1 gmunson gmunson    532 Nov 14 13:49 sampleProgramTwo.c
```

## File Systems

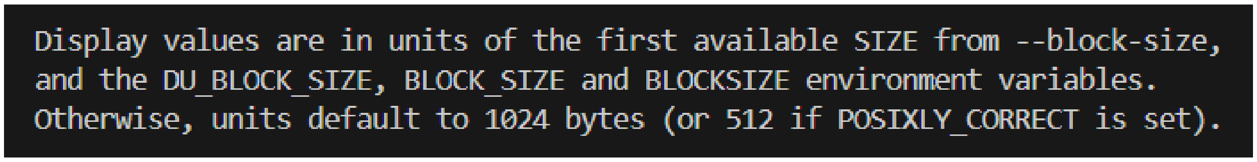
3. **Answer the following questions:**
   a) **Use du to report the usage of all the files in some of your directories (be sure to choose some with subdirectories), based on the order of information provided, which of the two tree traversal algorithms does du use?**



```
180K    /c/Users/gbaks/Downloads/cis452/assignments/assignment1-pointers
172K    /c/Users/gbaks/Downloads/cis452/assignments/assignment10-memory-segmentation
256K    /c/Users/gbaks/Downloads/cis452/assignments/assignment11-page-tables
224K    /c/Users/gbaks/Downloads/cis452/assignments/assignment12-virtual-memory
140K    /c/Users/gbaks/Downloads/cis452/assignments/assignment2-forks
88K     /c/Users/gbaks/Downloads/cis452/assignments/assignment3-pipes
908K    /c/Users/gbaks/Downloads/cis452/assignments/assignment4-threads
152K    /c/Users/gbaks/Downloads/cis452/assignments/assignment5-peterson
96K     /c/Users/gbaks/Downloads/cis452/assignments/assignment6-test-and-set
152K    /c/Users/gbaks/Downloads/cis452/assignments/assignment7-shared-memory
160K    /c/Users/gbaks/Downloads/cis452/assignments/assignment8-semaphores
200K    /c/Users/gbaks/Downloads/cis452/assignments/assignment9-scheduling
2.7M    /c/Users/gbaks/Downloads/cis452/assignments
430K    /c/Users/gbaks/Downloads/cis452/labs/lab1
340K    /c/Users/gbaks/Downloads/cis452/labs/lab10
353K    /c/Users/gbaks/Downloads/cis452/labs/lab11
523K    /c/Users/gbaks/Downloads/cis452/labs/lab2
389K    /c/Users/gbaks/Downloads/cis452/labs/lab3
376K    /c/Users/gbaks/Downloads/cis452/labs/lab4
380K    /c/Users/gbaks/Downloads/cis452/labs/lab5
304K    /c/Users/gbaks/Downloads/cis452/labs/lab6
389K    /c/Users/gbaks/Downloads/cis452/labs/lab7
1.8M    /c/Users/gbaks/Downloads/cis452/labs/lab8
429K    /c/Users/gbaks/Downloads/cis452/labs/lab9
5.6M    /c/Users/gbaks/Downloads/cis452/labs
4.7M    /c/Users/gbaks/Downloads/cis452/lectures/ch1-intro
1.7M    /c/Users/gbaks/Downloads/cis452/lectures/ch2-os-structures
2.8M    /c/Users/gbaks/Downloads/cis452/lectures/ch3-processes
1.4M    /c/Users/gbaks/Downloads/cis452/lectures/ch4-threads
1.1M    /c/Users/gbaks/Downloads/cis452/lectures/ch5-process-sync
1.4M    /c/Users/gbaks/Downloads/cis452/lectures/ch6-scheduling
3.0M    /c/Users/gbaks/Downloads/cis452/lectures/ch7-main-memory
4.3M    /c/Users/gbaks/Downloads/cis452/lectures/ch8-virtual-memory
21M     /c/Users/gbaks/Downloads/cis452/lectures
```

From examining the du output, it's clear that it uses a depth-first search (DFS) approach. In my directory structure, du fully lists each subdirectory and its contents within assignments before moving on to the labs directory, and it does the same for each lab and then each lecture in turn. This depth-first traversal is evident because it descends completely into each subdirectory, displaying everything within it, before moving to the next main directory, rather than listing all directories at each level first.

**b) What is the default block size used by du?**



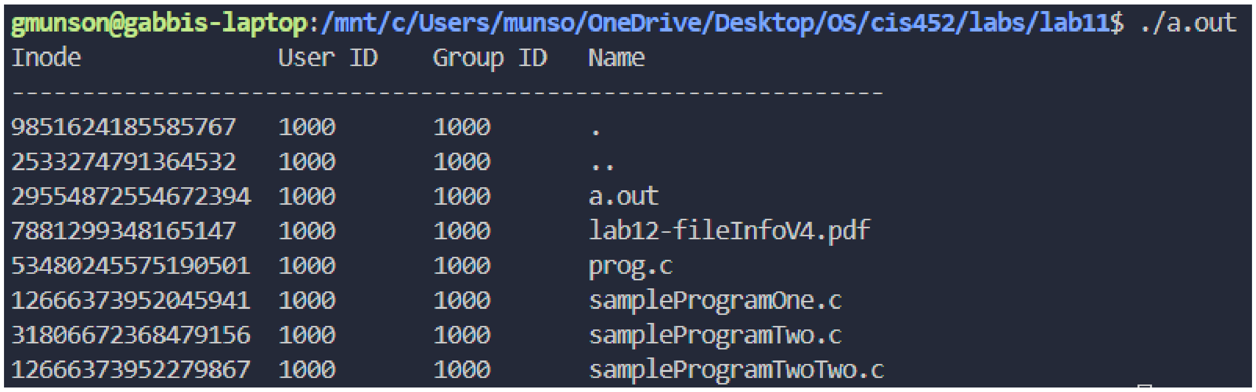
```
Display values are in units of the first available SIZE from --block-size,
and the DU_BLOCK_SIZE, BLOCK_SIZE and BLOCKSIZE environment variables.
Otherwise, units default to 1024 bytes (or 512 if POSIXLY_CORRECT is set).
```

Default block size is 1024 KB, or 1 KB which can be found be running *du –help*

**c) Speculate: given the intended purpose of du, why is the usage reported in blocks, instead of bytes?**

The *du* command reports disk usage in blocks instead of bytes because blocks provide a more practical and efficient measure of actual storage allocation. Since file systems allocate space in blocks, often in multiples of 1 KB (1024 bytes), this unit better reflects the disk space that files occupy. Reporting usage in blocks makes it faster to calculate and easier to interpret, especially for large files or directories. Using blocks instead of bytes also aligns with how the system manages disk space, giving me a more accurate view of my storage usage at the filesystem level.

## Programming Assignment



```
gmunson@gabbis-laptop:/mnt/c/Users/munso/OneDrive/Desktop/OS/cis452/labs/lab11$ ./a.out
Inode               User ID    Group ID    Name
------------------------------------------------------------
9851624185585767    1000       1000        .
2533274791364532    1000       1000        ..
29554872554672394   1000       1000        a.out
7881299348165147    1000       1000        lab12-fileInfoV4.pdf
53480245575190501   1000       1000        prog.c
12666373952045941   1000       1000        sampleProgramOne.c
31806672368479156   1000       1000        sampleProgramTwo.c
12666373952279867   1000       1000        sampleProgramTwoTwo.c
```