|  |
| --- |
| Getting Information about File Systems  과제물 4 |

학과 : 컴퓨터공학과

학번 : 2015313754

이름 : 길태형

목 차

[1. Statvfs를 이용한 File System 정보 출력 4](#_Toc39756631)

[1.1. Visual Studio Code에서 C프로그램 작성 및 실행 결과 4](#_Toc39756632)

[2. ass4.c프로그램 코드에 대한 상세 설명 5](#_Toc39756633)

[2.1. ass4.c 의 실행 흐름 5](#_Toc39756634)

[2.2. ass4.c 코드 내용 7](#_Toc39756635)

그림 목차

[[그림 1] Visual Studio Code에서 File System 정보 출력 프로그램 작성 4](#_Toc39756624)

[[그림 2] ass4프로그램 실행 결과와 df / command와의 비교. 5](#_Toc39756625)

[[그림 3] ass4.c의 코드 내용 6](#_Toc39756626)

[[그림 4] struct statvfs 사용 7](#_Toc39756627)

[[그림 5] file system의 정보 계산 7](#_Toc39756628)

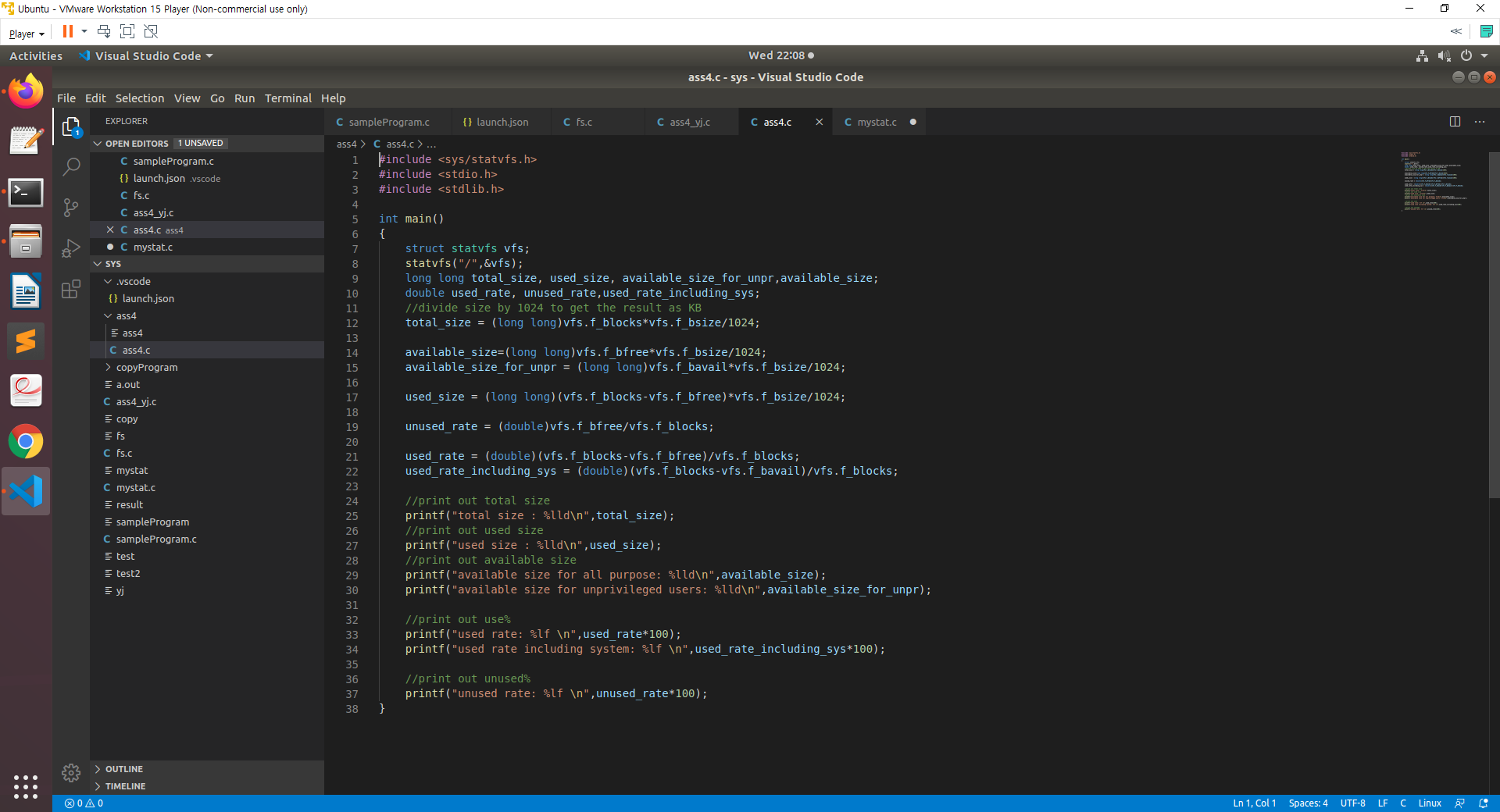
[[그림 6] 결과 출력 코드 8](#_Toc39756629)

[[그림 7] ass4 프로그램 실행 결과와 df / command와의 비교 8](#_Toc39756630)

1. Statvfs를 이용한 File System 정보 출력
   1. Visual Studio Code에서 C프로그램 작성 및 실행 결과

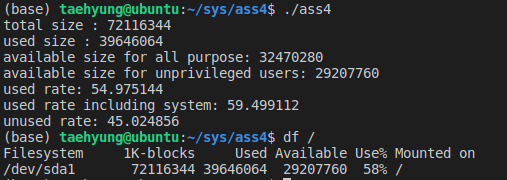
제 AWS의 반응속도가 너무 느려서, 개인 노트북의 VM의 linux환경에서 진행했습니다.

코드 내용은 아래와 같습니다.



[그림 1] Visual Studio Code에서 File System 정보 출력 프로그램 작성

Copy program을 작성한 후, Visual Studio Code의 Terminal에서 gcc 명령어를 활용하여 작성한 C 프로그램을 컴파일 했습니다. 작성한 C프로그램 파일은 ass4.c이고, 입력한 명령어는 ‘gcc ass4.c -o ass4’입니다. 결과로써, ass4 바이너리 파일을 산출했습니다.



[그림 2] ass4프로그램 실행 결과와 df / command와의 비교.

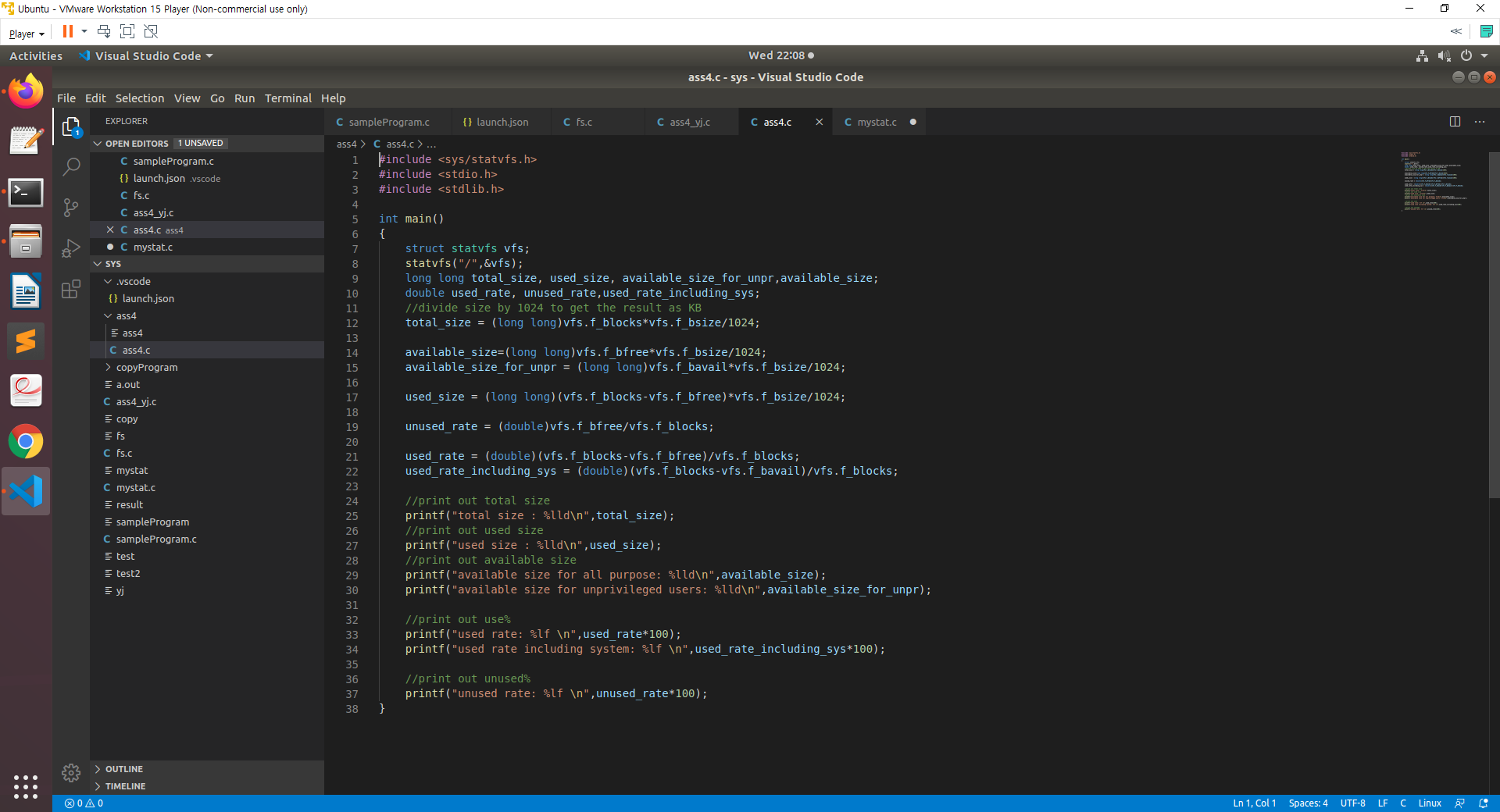
Size의 단위는 KB입니다. Available size의 경우, statvfs의 f\_bavail member를 사용하여 unprivileged user만을 위한 여유공간을 구할 수도 있고, statvfs의 f\_bfree member를 사용하여 unprivileged user가 사용할 수 있는 여유공간과, unprivileged user가 사용할 수 없는 여유공간을 합한 총 여유공간을 구할 수도 있습니다.

Rate의 단위는 %이고, used rate의 경우, f\_bvail에 해당하는 여유공간과, f\_bfree에 해당하는 여유공간 모두 %로 표현했습니다.

ass4 프로그램 실행 결과와 df / command를 비교했습니다. Total size와 used size는 차이가 없지만, df / command의 available size는 unprivileged user가 사용할 수 있는 것만을 나타냈고, use% 계산값에서 ass4 프로그램과 약간의 차이가 있었습니다.

1. ass4.c프로그램 코드에 대한 상세 설명
   1. ass4.c 의 실행 흐름

아래 그림은 ass4.c의 코드 내용입니다.



[그림 3] ass4.c의 코드 내용

ass4.c 코드는

1. ‘/’ 경로의 file system 정보 저장

2. total size, used size, available size, use%, unuse% 계산

3. 계산한 값들의 출력

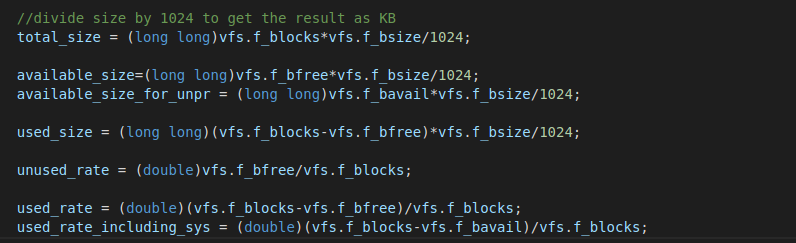
의 흐름으로 구성되어 있습니다.

* 1. ass4.c 코드 내용



[그림 4] struct statvfs 사용

Struct statvfs type의 변수를 선언하여, statvfs()함수를 사용하여 ‘/’경로의 file system 정보를 vfs에 저장합니다.



[그림 5] file system의 정보 계산

vfs변수에 저장된 값을 이용하여 file system의 정보를 계산합니다.

계산할 정보들은 total size, available size, used size, use%, unuse%입니다.

-total size

File system의 total size는 (file system의 총 block의 개수) \* (block 1개의 size)입니다.

Block size의 단위는 Byte이므로, KB 단위로 나타내기 위해 1024로 나눈 값을 저장했습니다.

(file system의 총 block의 개수)는 f\_blocks member에 저장되어 있고, block 1개의 size는 f\_bsize에 저장되어 있습니다. 따라서, 계산식은

Vfs.f\_blocks \* vfs.f\_bsize /1024가 됩니다. 계산 결과값을 total\_size (long long int)변수에 저장했습니다.

-available size

File system의 여유공간과 관계 있는 vfs의 멤버는 f\_bavail과 f\_bfree가 있습니다.

f\_bavail은 unprivileged user가 사용할 수 있는 여유 block의 개수를 가지고 있고, f\_bfree는 f\_bavail에 저장된 block의 개수에, unprivileged user가 사용할 수 없는 여유 block의 개수까지 합해서 저장하고 있습니다.

따라서, 계산식은 vfs.f\_bfree\*vfs.f\_bsize/1024와 vfs.f\_bavail\*vfs.f\_bsize/1024가 됩니다.

-used size

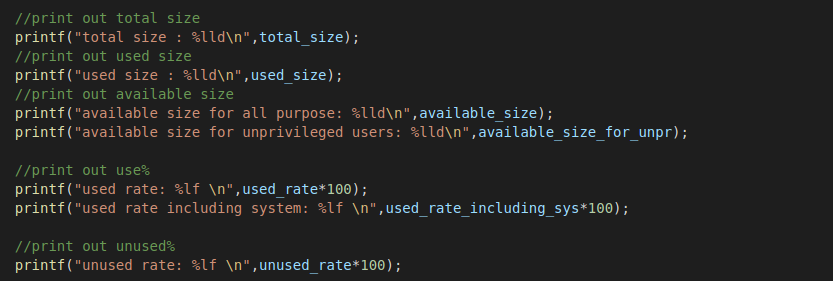
Used size는 total size에서 available size를 빼서 계산합니다. 여기서, available size는 f\_bavail을 통해 구한 값이 아닌, vfs.f\_bfree를 통해 구한 값을 빼야 unprivileged user가 사용하지는 못하지만 아직 사용되지 않은 여유공간을 포함해서 뺄 수 있습니다.

따라서 계산식은, (vfs.f\_blocks-vfs.f\_bfree)\*vfs.f\_bsize/1024가 됩니다.

-rate

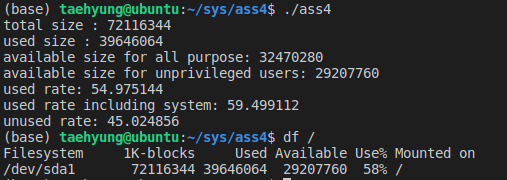
Unused\_rate, used\_rate는 각 해당하는 block의 개수를 File System의 전체 block의 개수로 나눠서 계산했습니다.

Unused\_rate의 경우, 해당하는 block의 개수는 f\_bfree입니다. 이 File system의 전체 block인 f\_blocks로 나눠서 계산했습니다.



[그림 6] 결과 출력 코드

Printf 함수를 이용하여 계산한 값들을 출력했습니다.



[그림 7] ass4 프로그램 실행 결과와 df / command와의 비교