

[운영체제 과제1] 파일이나 디렉토리의 용량 계산

2021.9.20.(월)

1. 과제 목표

- Unix에서 디렉토리 용량 계산 작업을 해 봄으로써 파일 정보 및 파일 시스템 관련한 Unix 시스템 프로그래밍 능력을 제고한다.

2. 과제

1)과제 내용

```
-bash-3.2$ ls -l main
-rwxr-xr-x  1 engineerjkk0510 2016      6556 10월  3일  13:23 main
-bash-3.2$ du -h main
7K  main
-bash-3.2$ ls -l main
-rwxr-xr-x  1 engineerjkk0510 2016      6556 10월  3일  13:23 main
```

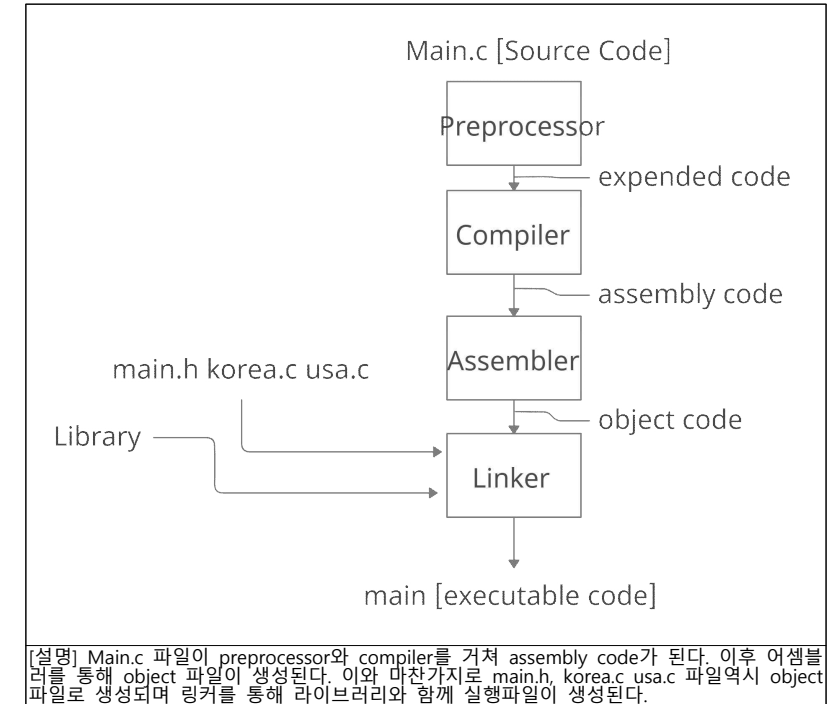
[설명] Makefile로부터 생성된 실행 파일의 실제 크기는 6556바이트이지만, du -h가 용량 계산시 이를 7KB 단위로 변환하여 계산한다. 즉 파일 크기를 1024의 floor 함수로 계산한 후, +1한다. $\text{floor}(6556/1024)+1=6+1=7\text{KB}$

```
-bash-3.2$ du -h junekoo
11K  junekoo
-bash-3.2$ cd junekoo
-bash-3.2$ du -h
11K  .
-bash-3.2$ ls
a.out  cs1.txt  cs2.txt  hello.c
-bash-3.2$ du -h a.out cs1.txt cs2.txt hello.c
7K  a.out
1K  cs1.txt
1K  cs2.txt
1K  hello.c
```

[설명] 빈 디렉토리는 .와 ..만을 포함하는데, 이들 용량을 1K로 간주한다. .과 ..의 용량은 현 디렉토리 용량 계산시 포함하지 않는다. 위 그림에서 보면 'junekoo'폴더의 용량은 11K이지만 junekoo내부에 파일을 전체 합은 $7\text{K}+1\text{K}+1\text{K}+1\text{K}=10\text{K}$ 이다. 따라서 빈 디렉토리가 1K임을 알 수 있다.

2) 문제 해결 방법

과제에 주어진 요구사항을 만족하고 용량을 계산하는 조건을 만족하는 것에 집중하였다. 따라서 내부 C코드는 누구나 보아도 쉽게 이해가능한 가독성이 높은 간단한 printf로 구성된 예제를 작성하였다. 이후 요구하는 du -h를 통해 파일과 디렉토리 용량을 계산한다.



[설명] Main.c 파일이 preprocessor와 compiler를 거쳐 assembly code가 된다. 이후 어셈블러를 통해 object 파일이 생성된다. 이와 마찬가지로 main.h, korea.c, usa.c 파일역시 object 파일로 생성되며 링커를 통해 라이브러리와 함께 실행파일이 생성된다.

3) 프로그램 구조

첫 번째. main.c 파일을 생성한다.

```
#include "main.h"

int main()
{
    printf("Hello, I'am a main function\n");

    proc_korea();
    proc_usa();

    return 0;
}
```

[설명] main.c 파일로 본인의 printf 실행 후 각각의 함수를 실행한다.

두 번째. 각각의 함수가 담긴 c파일 생성 및 헤더파일 생성

```
#include <stdio.h>

void proc_korea();
void proc_usa();
```

[설명] main.h 파일로 stdio.h를 include하고 각각의 함수들이 담겨있다.

```
#include "main.h"

void proc_korea()
{
    printf("Hello, I'm Korean\n");
}
```

[설명] korea.c 파일로 Hello, I'm Korean을 출력한다.

```
#include "main.h"

void proc_usa()
{
    printf("Hello, I'm American\n");
}
```

[설명] usa.c 파일로 Hello, I'm American을 출력한다.

세 번째. Makefile 생성

```
main: main.o korea.o usa.o
    gcc -o main main.o korea.o usa.o
main.o: main.c main.h
    gcc -c main.c
korea.o: korea.c main.h
    gcc -c korea.c
usa.o: usa.c main.h
    gcc -c usa.c
```

[설명] 각각의 생성된 object 파일을 통해 실행파일을 생성하는 Makefile이다.

네 번째. 실행파일 생성

```
-bash-3.2$ make
gcc -c main.c
gcc -c korea.c
gcc -c usa.c
gcc -o main main.o korea.o usa.o
-bash-3.2$ ls
3rd                kor.o
8_15.c             local.cshrc
8_9.c              local.login
america.c           local.profile
copy.c              longest
copy.h              longest.c
copy.o              main
```

[설명] make 명령어를 통해 object 파일을 생성 후 오른쪽 가장 하단에 main 실행파일이 생성됨을 확인할 수 있다.

다섯 번째. 파일과 디렉토리 용량 계산

```
-bash-3.2$ ls -l main
-rwxr-xr-x  1 engineerjkk0510 2016      6556 10월  3일  13:23 main
-bash-3.2$ du -h main
7K  main
-bash-3.2$ ls -l main
-rwxr-xr-x  1 engineerjkk0510 2016      6556 10월  3일  13:23 main
```

[설명] Makefile로부터 생성된 실행 파일의 실제 크기는 6556바이트이지만, du -h가 용량 계산시 이를 7KB 단위로 변환하여 계산한다. 즉 파일 크기를 1024의 floor 함수로 계산한 후, +1한다. floor(6556/1024)+1=6+1=7KB

```
~$ du -h
-bash-3.2$ du -h junekoo
11K  junekoo
-bash-3.2$ cd junekoo
-bash-3.2$ du -h
11K  .
-bash-3.2$ ls
a.out  cs1.txt  cs2.txt  hello.c
-bash-3.2$ du -h a.out cs1.txt cs2.txt hello.c
7K  a.out
1K  cs1.txt
1K  cs2.txt
1K  hello.c
```

[설명] 빈 디렉토리는 .와 ..만을 포함하는데, 이들 용량을 1K로 간주한다. '과 ..'의 용량은 현 디렉토리 용량 계산시 포함하지 않는다. 위 그림에서 보면 'junekoo'폴더의 용량은 11K이지만 junekoo내부에 파일을 전체 합은 7K+1K+1K+1K=10K이다. 따라서 빈 디렉토리가 1K임을 알 수 있다.

여섯 번째 : 전체적인 파일 리스트

```
-bash-3.2$ ls -l main.c main.h korea.c usa.c Makefile main
-rwxr-xr-x  1 engineerjkk0510 2016      73 10월  3일  02:23 korea.c
-rwxr-xr-x  1 engineerjkk0510 2016     6556 10월  3일  14:28 main
-rwxr-xr-x  1 engineerjkk0510 2016     120 10월  3일  13:23 main.c
-rw-r--r--  1 engineerjkk0510 2016      59 10월  3일  02:57 main.h
-rwxr-xr-x  1 engineerjkk0510 2016     173 10월  3일  13:08 Makefile
-rwxr-xr-x  1 engineerjkk0510 2016      73 10월  3일  02:22 usa.c
```

[설명] 프로그램 구조는 main.c, korea.c, usa.c과 함께 헤더파일인 main.h로 구성된다. Makefile의 내부구조를 통하여 'main' 실행파일을 생성하게 된다.

4) 실행 결과

```
-bash-3.2$ main
Hello, I'am a main function
Hello, I'm Korean
Hello, I'm American
```

[설명] Makefile을 통해 생성된 실행파일 main을 실행하면 각각의 함수들이 호출되어 문장을 프린트함을 확인할 수 있다.

4) 결론

리눅스 프로그래밍이 생소하여 과제를 이해하는데에도 상당한 시간이 걸렸습니다. 하지만 차근차근 교재를 읽고 강의를 반복해서 학습하며 무엇을 의도하는지 파악할 수 있었습니다.

du -h 명령어부터 Makefile 생성하는 방법, 컴파일의 내부구조 등 다방면으로 학습할 수 있었던 좋은 과제였습니다. 이번 과제를 수행하면서 리눅스 프로그래밍에 입문할수있게되어 이러한 기획으로 과제를 내주신 교수님께 진심으로 감사드립니다.

5) 부록 및 참고문헌

유튜브 :

<https://www.youtube.com/watch?v=jnJL6ppn26Q>

pdf :

[http://ebook.pldworld.com/_eBook/make/make_utility_lecture_\(cWyANG\).pdf](http://ebook.pldworld.com/_eBook/make/make_utility_lecture_(cWyANG).pdf)

네이버 블로그 :

<https://m.blog.naver.com/PostView.naver?isHttpsRedirect=true&blogId=jsky10503&logNo=220734514135>

StackOverflow :

<https://stackoverflow.com/questions/45521964/make-fatal-error-command-failed-for-target>