

Package Manager

Package 는 흔히 알고 있는 App, Program 이라고 할 수 있다.

이러한 package 를 다루는 작업을 편리하고 안전하게 수행하기 위해 사용되는 툴을 package manager 라고 한다.

apt-get command not found

amazon aws 에서는 apt-get 지원하지 않는다. 따라서 yum 을 대신 사용하면 해결 가능하다.

sudo yum update 를 통해 package 를 update 한다.

```
ec2-user:~/environment $ sudo yum update
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
233 packages excluded due to repository priority protections
Resolving Dependencies
--> Running transaction check
--> Package aws-cfn-bootstrap.noarch 0:2.0-9.amzn2 will be updated
--> Package aws-cfn-bootstrap.noarch 0:2.0-10.amzn2 will be an update
--> Package cloud-init.noarch 0:19.3-44.amzn2 will be updated
--> Package cloud-init.noarch 0:19.3-45.amzn2 will be an update
--> Package containerd.x86_64 0:1.4.6-7.amzn2 will be updated
--> Package containerd.x86_64 0:1.4.6-8.amzn2 will be an update
--> Package ec2-utils.noarch 0:1.2-46.amzn2 will be updated
--> Package ec2-utils.noarch 0:1.2-47.amzn2 will be an update
--> Package terraform.x86_64 0:1.1.6-1 will be updated
--> Package terraform.x86_64 0:1.1.7-1 will be an update
--> Package tzdata.noarch 0:2021a-1.amzn2 will be updated
--> Package tzdata.noarch 0:2021e-1.amzn2 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package                                Arch                                Version                                Repository                                Size
=====
Updating:
aws-cfn-bootstrap                      noarch                              2.0-10.amzn2                          amzn2-core                               768 k
cloud-init                             noarch                              19.3-45.amzn2                         amzn2-core                               926 k
containerd                             x86_64                              1.4.6-8.amzn2                         amzn2extra-docker                       24 M
=====
```

top 은 컴퓨터의 작업관리자와 같이 현재 실행되고 있는 파일들을 보여준다.

```
Tasks: 110 total, 2 running, 67 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni,100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 1005824 total, 372460 free, 327660 used, 305704 buff/cache
KiB Swap: 499996 total, 478492 free, 21504 used, 537296 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
866	root	20	0	0	0	0	S	0.0	0.0	0:00.00	xenbus
867	root	20	0	0	0	0	S	0.0	0.0	0:00.01	xenwatch
889	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	nme-wq
918	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	ipv6_addrconf
923	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kstrp
971	root	20	0	0	0	0	I	0.0	0.0	0:00.08	kworker/0:2
1782	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	ata_sff
1790	root	20	0	0	0	0	S	0.0	0.0	0:00.00	scsi_eh_0
1799	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	scsi_tmf_0
1803	root	20	0	0	0	0	S	0.0	0.0	0:00.00	scsi_eh_1
1805	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	scsi_tmf_1
1887	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	xfs-buf/xvda1
1888	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	xfs-data/xvda1
1889	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	xfs-conv/xvda1
1890	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	xfs-cil/xvda1
1891	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	xfs-reclaim/xvd
1892	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	xfs-log/xvda1
1893	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	xfs-eofblocks/x
1894	root	20	0	0	0	0	S	0.0	0.0	0:00.25	xfsaild/xvda1
1896	root	0	-20	0	0	0	I	0.0	0.0	0:00.04	kworker/0:1H
1958	root	20	0	55776	10228	9860	S	0.0	1.0	0:00.23	systemd-journal
2082	root	20	0	116752	1448	1196	S	0.0	0.1	0:00.00	lvmtd

이 때 htop 을 따로 설치해서 top 보다 더 나은 monitoring tool 을 생성하자.

sudo yum install htop 을 통해 htop 을 설치하고 실행 시킨 모습이다.

```
CPU[|||||||||||||||||||||||||||||||||||||||||] 0.7% Tasks: 50, 00 thr; 1 running
Mem[||||||||||||||||||||||||||||||||||||||||] 319M/982M Load average: 0.01 0.07 0.08
Swap[||||] 24.2M/488M Uptime: 00:22:46
```

PID	USER	PRI	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	Command
3448	root	20	0	121M	3052	2700	S	0.7	0.3	0:00.80	/bin/bash /usr/bin/logd-j-cve-2021-44228-hotpatch -w 1800 -m 10
12946	ec2-user	20	0	147M	4724	2936	S	0.7	0.5	0:00.27	sshd: ec2-user@notttyerval:5000,"nodePath":"/home/ec2-user/.c9/node_modules","tmuxBin":"/home/ec2-user/.c9/bin/tmux","root"
17011	ec2-user	20	0	126M	3968	3164	R	0.7	0.4	0:00.02	htop/lib/systemd/systemd --switched-root --system --deserialize 22
6158	ec2-user	20	0	123M	2964	2180	S	0.0	0.3	0:00.48	/home/ec2-user/.c9/bin/tmux -u2 -L cloud92.2 new -s cloud9_terminal_162 export ISOUTPUTPANE=0;bash -l ; set -q -g status of
3265	root	16	-4	57688	1636	1252	S	0.0	0.2	0:00.00	/sbin/auditd
3262	root	16	-4	57688	1636	1252	S	0.0	0.2	0:00.01	/sbin/auditd
3326	dbus	20	0	60352	3348	4092	S	0.0	0.3	0:00.22	/usr/bin/dbus-daemon --system --address=systemd: --nofork --nopidfile --systemd-activation
3395	rpc	20	0	69348	2612	2212	S	0.0	0.3	0:00.01	/sbin/rpcbind -w
3423	root	20	0	197M	2240	1916	S	0.0	0.2	0:00.00	/usr/sbin/gssproxy -D
3424	root	20	0	197M	2240	1916	S	0.0	0.2	0:00.00	/usr/sbin/gssproxy -D
3425	root	20	0	197M	2240	1916	S	0.0	0.2	0:00.00	/usr/sbin/gssproxy -D
3426	root	20	0	197M	2240	1916	S	0.0	0.2	0:00.00	/usr/sbin/gssproxy -D
3427	root	20	0	197M	2240	1916	S	0.0	0.2	0:00.00	/usr/sbin/gssproxy -D
3406	root	20	0	197M	2240	1916	S	0.0	0.2	0:00.00	/usr/sbin/gssproxy -D
3419	libstorag	20	0	12600	1896	1732	S	0.0	0.2	0:00.01	/usr/bin/lsm -d
3444	root	20	0	2452	2576	2280	S	0.0	0.3	0:00.00	/usr/lib/systemd/systemd-logind
3448	root	20	0	121M	3052	2700	S	0.0	0.3	0:00.79	/bin/bash /usr/bin/logd-j-cve-2021-44228-hotpatch -w 1800 -m 10
3595	chrony	20	0	117M	3084	2712	S	0.0	0.3	0:00.03	/usr/sbin/chronyd
3613	rngd	20	0	94096	2236	1712	S	0.0	0.2	0:00.05	/sbin/rngd -f --fill-watermark=0 --exclude-jitter
3952	root	20	0	90M	1256	1252	S	0.0	0.1	0:00.00	/sbin/dhclient -q -lf /var/lib/dhclient/dhclient-eth0.lease -pf /var/run/dhclient-eth0.pid -H ip-172-31-16-100 eth0
4092	root	20	0	90M	1248	1460	S	0.0	0.2	0:00.00	/sbin/dhclient -6 -nw -lf /var/lib/dhclient/dhclient6-eth0.lease -pf /var/run/dhclient6-eth0.pid eth0 -H ip-172-31-16-100
4595	chrony	20	0	117M	3084	2712	S	0.0	0.2	0:00.03	/usr/sbin/chronyd

훨씬 더 성능이 개선되었음을 확인할 수 있다.

다시 htop 을 삭제하기 위해선 `sudo yum remove htop` 을 통해 간단하게 삭제가 가능하다.

```
Dependencies Resolved

-----
Package                               Arch              Version           Repository        Size
-----
Removing:
htop                                  x86_64            2.0.2-1.amzn2.0.2 @amzn2-core      207 k
-----

Transaction Summary
-----
Remove 1 Package

Installed size: 207 k
Is this ok [y/N]: y
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Erasing   : htop-2.0.2-1.amzn2.0.2.x86_64                1/1
Verifying : htop-2.0.2-1.amzn2.0.2.x86_64                1/1

Removed:
htop.x86_64 0:2.0.2-1.amzn2.0.2

Complete!
```

정리

1. `sudo yum update` : 최신상태로 목록을 유지한다. upgrade 와는 다름.
2. `sudo yum search htop` : 원하는 package 를 찾는다.
3. `sudo yum install htop` : 설치한다.
4. `sudo yum upgrade htop` : 업그레이드한다.
5. `sudo yum remove htop` : 삭제한다.

wget

wget 이라는 프로그램을 이용해서 url 이라는 파일을 다운받을 수 있다.

wget 뒤에 url 링크를 복사하면 쉽게 다운로드 할 수 있다.

```
ec2-user:~/environment $ wget https://unsplash.com/photos/yihlaRCCvd4/download?ixid=MnwxdMjA3fDB8MXczZWFiY2h8Mnx8ZG9nFGVufDB8fhx8MTY0NjM4Mdc1OA&force=true
[1] 19328
ec2-user:~/environment $ --2022-03-04 10:52:30-- https://unsplash.com/photos/yihlaRCCvd4/download?ixid=MnwxdMjA3fDB8MXczZWFiY2h8Mnx8ZG9nFGVufDB8fhx8MTY0NjM4Mdc1OA
Resolving unsplash.com (unsplash.com)... 151.101.1.181, 151.101.65.181, 151.101.129.181, ...
Connecting to unsplash.com (unsplash.com)[151.101.1.181]:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://images.unsplash.com/photo-1530281700549-e82e7bf110d6?ixlib=rb-1.2.1&q=80&fm=jpg&crop=entropy&cs=tinysrgb [following]
--2022-03-04 10:52:30-- https://images.unsplash.com/photo-1530281700549-e82e7bf110d6?ixlib=rb-1.2.1&q=80&fm=jpg&crop=entropy&cs=tinysrgb
Resolving images.unsplash.com (images.unsplash.com)... 146.75.34.208, 2a04:4e42:78::720
Connecting to images.unsplash.com (images.unsplash.com)[146.75.34.208]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2156094 (2.1M) [image/jpeg]
Saving to: 'download?ixid=MnwxdMjA3fDB8MXczZWFiY2h8Mnx8ZG9nFGVufDB8fhx8MTY0NjM4Mdc1OA'

100%[----->] 2,156,094  --K/s  in 0.01s

2022-03-04 10:52:30 (167 MB/s) - 'download?ixid=MnwxdMjA3fDB8MXczZWFiY2h8Mnx8ZG9nFGVufDB8fhx8MTY0NjM4Mdc1OA' saved [2156094/2156094]
```

파일을 다운 받을 때 이름을 수정하는 방법은 `wget -O` (수정하고자 하는 이름) (파일링크) 이다.

```
ec2-user:~/environment $ wget -O dog.jpeg https://unsplash.com/photos/yihlaRCCvd4/download?ixid=MmwxMjA3fDB8MDkxZWYyZ2h8Mnx8ZG9nfGVuFDB8fHx8MTY0NjM4Mdc10A&force=true
[1] 20743
--2022-03-04 10:55:54-- https://unsplash.com/photos/yihlaRCCvd4/download?ixid=MmwxMjA3fDB8MDkxZWYyZ2h8Mnx8ZG9nfGVuFDB8fHx8MTY0NjM4Mdc10A
Resolving unsplash.com (unsplash.com)... ec2-user:~/environment $ 151.101.193.181, 151.101.1.181, 151.101.65.181, ...
Connecting to unsplash.com (unsplash.com)[151.101.193.181]:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://images.unsplash.com/photo-1530281700549-e82e7bf110d6?ixlib=rb-1.2.1&q=80&fm=jpg&crop=entropy&cs=tinsyrbg [following]
--2022-03-04 10:55:54-- https://images.unsplash.com/photo-1530281700549-e82e7bf110d6?ixlib=rb-1.2.1&q=80&fm=jpg&crop=entropy&cs=tinsyrbg
Resolving images.unsplash.com (images.unsplash.com)... 199.232.38.208, 2a04:4e42:78::720
Connecting to images.unsplash.com (images.unsplash.com)[199.232.38.208]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2156094 (2.1M) [image/jpeg]
Saving to: 'dog.jpeg'

100%[=====] 2,156,094  --K/s  in 0.04s

2022-03-04 10:55:54 (50.3 MB/s) - 'dog.jpeg' saved [2156094/2156094]

[1]+  Done                  wget -O dog.jpeg https://unsplash.com/photos/yihlaRCCvd4/download?ixid=MmwxMjA3fDB8MDkxZWYyZ2h8Mnx8ZG9nfGVuFDB8fHx8MTY0NjM4Mdc10A
```

`wget -O dog.jpeg` 링크 로 파일을 저장하였고 `ls -l` 을 통해 파일이 저장된 것을 확인할 수 있다.

```
ec2-user:~/environment $ ls -l
total 4220
-rw-rw-r-- 1 ec2-user ec2-user 2156094 Mar  2 16:12 dog.jpeg
-rw-rw-r-- 1 ec2-user ec2-user 2156094 Mar  2 16:12 hello.jpeg
-rw-r--r-- 1 ec2-user ec2-user    569 Feb 15 03:57 README.md
```

Git

버전 관리 시스템이라고 하는 카테고리에 속하는 구체적인 제품 중의 하나이다.

git 으로 github 소스 코드를 다운로드 하는 방법

1. github 의 clone 의 https 를 copy 한다.
2. git clone (github 의 https) (특정 파일) 을 입력하고 특정 파일을 다운로드 받는다.

```
ec2-user:~/environment $ git clone https://github.com/facebook/react.git react_src
Cloning into 'react_src'...
remote: Enumerating objects: 201184, done.
remote: Total 201184 (delta 0), reused 0 (delta 0), pack-reused 201184
Receiving objects: 100% (201184/201184), 172.09 MiB | 24.94 MiB/s, done.
Resolving deltas: 100% (142830/142830), done.
ec2-user:~/environment $ ls -l
total 4228
-rw-rw-r-- 1 ec2-user ec2-user 2156094 Mar  2 16:12 dog.jpeg
-rw-rw-r-- 1 ec2-user ec2-user 2156094 Mar  2 16:12 hello.jpeg
drwxrwxr-x 9 ec2-user ec2-user  4096 Mar  4 12:24 react
drwxrwxr-x 9 ec2-user ec2-user  4096 Mar  4 12:25 react_src
-rw-r--r-- 1 ec2-user ec2-user   569 Feb 15 03:57 README.md
```

CLI

명령어로 컴퓨터를 제어하는 방식은 하나는 GUI(Graphical user interface)가 있고, CLI(Command line interface)가 있다.

CLI 방식이 훨씬 적은 '컴퓨터 파워'를 차지하면서 컴퓨터를 동작시킬 수 있는 방법이다. Heavy 한 일을 하기 위해선 CLI 방식이 필수이다.

또한, GUI 방식은 쉽다는 장점이 있지만, 상당히 많은 노동이 필요하다. 순차적으로 진행되는 방식을 하기엔 어려움이 있다. 반면 CLI는 일이 진행되는 과정을 확인할 필요없이, 결과값만 받으면 되기 때문에 편리하다.

CLI를 사용했을 때 얻을 수 있는 효과는 무엇일까?

어떤 명령을 실행했을 때 그 프로그램이 끝나는 것을 확인한 후에 다음 프로그램을 실행하는 것이 아니라, 순차적으로 내가 해야 될 일을 쭉 적어서 컴퓨터에게 보내주면 최종적인 결과만을 알려준다. 이는 명령을 통해 컴퓨터를 제어할 때 굉장히 중요한 요소이고, 컴퓨터 프로그래밍을 배우는 이유이다.

pipe

리눅스는 하나의 명령의 실행 결과를 다른 명령의 입력으로 주는 기능을 가지고 있으며 이를 pipe 라고 한다.

nano 를 이용해 linux.txt 라는 파일을 생성하고 text 를 넣은 후 cat 을 통해 linux.txt 를 확인한다.

```
ec2-user:~/environment $ cat linux.txt
```

grep 은 파일 안에서 특정 텍스트가 포함된 행만을 출력하고 싶을 때 사용하는 기능이다.

사용법 : grep (특정 텍스트) (텍스트 파일)

```
ec2-user:~/environment $ grep linux linux.txt
소프트웨어 라이브러리: 실행 중인 프로세스가 사용할 수 있는 코드를 포함한다. ELF 포맷 실행 파일을 사용하는 리눅스 시스템에서 동적 라이브러리 이용을 관리하는 동적 링커의 이름은 ld-linux.so이다. 사용자들이 직접 소프트웨어를 컴파일할 수 있도록 시스템이 구성되어 있다면 헤더 파일 또한 설치된 라이브러리의 인터페이스를 기술하기 위해 포함된다. 리눅스 시스템에서 가장 흔히 사용되는 소프트웨어 라이브러리 GNU C 라이브러리(glibc) 외에 SDL, 메사 등 그 밖의 수많은 라이브러리들이 존재한다.
C 표준 라이브러리는 컴퓨터 시스템에서 C 프로그램을 실행하기 위해 필요한 라이브러리이며, 표준이 되는 GNU C 라이브러리라 할 수 있다. 임베디드 시스템의 경우 musl, EGLIBC(데비안에 사용되는 glibc 포크), uClibc(uClinux용으로 설계됨) 등 다른 대안들이 개발되었으나 마지막 2개는 더 이상 유지보수되지 않는다. 안드로이드는 Bionic이라는 자체 C 라이브러리를 사용한다.
리눅스 커널은 널리 이식된 운영 체제 커널로서, 휴대 전화에서부터 슈퍼컴퓨터에 이르기까지 여러 장치에서 이용이 가능하다. 매우 다양한 범위의 컴퓨터 아키텍처 위에서 동작하는데, 여기에는 휴대용 ARM 기반 iPAQ와 IBM 메인프레임 시스템 z9이나 시스템 z10이 포함된다.[75] 주류에서 조금 벗어난 아키텍처를 위한 특수한 배포판과 커널 포크(fork)들이 존재한다. 예를 들어 ELKS 커널 포크는 인텔 8086이나 인텔 80286 16비트 마이크로프로세서에서 동작하는 반면 µClinux 커널 포크는 메모리 관리 장치(MPU)가 없는 시스템에서 동작할 수 있다. 커널은 또한 메인토시 컴퓨터(파워PC와 인텔 프로세서 모두), PDA, 비디오 게임기, 휴대용 음악 플레이어, 휴대 전화 등에서 볼 수 있듯이 제조사가 만든 운영 체제의 이용을 염두에 둔 아키텍처 위에서 동작하기도 한다.
"Re: How to pronounce Linux?". 뉴스그룹: comp.os.linux. 1992년 4월 23일. Usenet: 1992Apr23.123216.22024@klaava.Helsinki.FI. 2007년 1월 9일에 확인함.
Eric Brown (2019년 3월 29일). "Linux continues advance in smart TV market". 《linuxgizmos.com》. 2020년 5월 15일에 확인함.
Linus Torvalds (1992년 10월 14일). "Chicken and egg: How was the first linux gcc binary created?.". 뉴스그룹: comp.os.minix. Usenet: 1992Oct12.100843.26287@klaava.Helsinki.FI. 2013년 8월 17일에 확인함.
Torvalds, Linus (March 1994). "Index of /pub/linux/kernel/SillySounds". October 8, 2009에 원본 문서에서 보존된 문서. August 3, 2009에 확인함.
"Linux TV: Television with Linux". linuxtv.org. November 6, 2013에 원본 문서에서 보존된 문서. October 16, 2013에 확인함.
"Part I. Video for Linux Two API Specification". 《Chapter 7. Changes》. linuxtv.org. October 17, 2013에 원본 문서에서 보존된 문서. October 16, 2013에 확인함.
"Introducing sinhala linux". 2015년 11월 24일에 원본 문서에서 보존된 문서. 2015년 11월 23일에 확인함.
Andrew D. Balsa; Coauthors. "The linux-kernel mailing list FAQ". 《The Linux Kernel Archives》. Kernel.org. 2012년 10월 1일에 원본 문서에서 보존된 문서. 2013년 6월 13일에 확인함. ...we have tried to use the word "Linux" or the expression "Linux kernel" to designate the kernel, and GNU/Linux to designate the entire body of GNU/GPL'ed OS software,... ...many people forget that the linux kernel mailing list is a forum for discussion of kernel-related matters, not GNU/Linux in general...
```

이 때 특정 program 에서 텍스트를 찾기 위해서 pipe 를 이용할 수 있다.

사용법 : program |(shift+) grep (텍스트)

ex) ls --help | grep sort

```
ec2-user:~/environment $ ls --help | grep sort
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.
  -c              with -lt: sort by, and show, ctime (time of last
                  with -l: show ctime and sort by name;
                  otherwise: sort by ctime, newest first
  -f              do not sort, enable -aU, disable -ls --color
                  can be augmented with a --sort option, but any
                  use of --sort=none (-U) disables grouping
  -r, --reverse   reverse order while sorting
  -S              sort by file size
  --sort=WORD      sort by WORD instead of name: none (-U), size (-S),
                  as sort key if --sort=time
  -t              sort by modification time, newest first
  -u              with -lt: sort by, and show, access time;
                  with -l: show access time and sort by name;
                  otherwise: sort by access time
  -U              do not sort; list entries in directory order
  -v              natural sort of (version) numbers within text
  -X              sort alphabetically by entry extension
```

ex) ps aux | grep apache

```
ec2-user:~/environment $ ps aux | grep apache
ec2-user 24370  0.0  0.0 119420  964 pts/1    S+   13:02   0:00 grep --color=auto apache
```

IO redirection

IO : I 는 input, O 는 output 을 의미한다.

redirection : 방향을 바꾸는 것을 의미한다.

Output

사용법 : program > 파일

ex) ls -l > result.txt

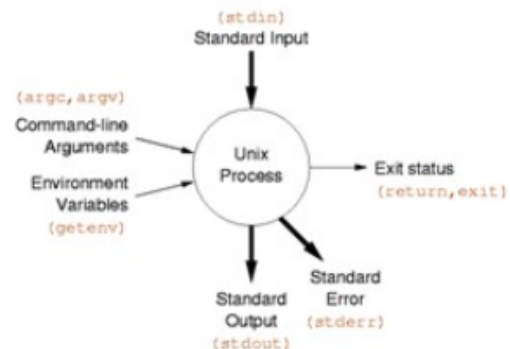
```
ec2-user:~/environment $ ls -l > result.txt
ec2-user:~/environment $ cat result.txt
total 4328
-rw-rw-r-- 1 ec2-user ec2-user 2156094 Mar  2 16:12 dog.jpeg
-rw-rw-r-- 1 ec2-user ec2-user 2156094 Mar  2 16:12 hello.jpeg
-rw-rw-r-- 1 ec2-user ec2-user  99501 Mar  4 12:41 linux.txt
drwxrwxr-x 9 ec2-user ec2-user  4096 Mar  4 12:24 react
drwxrwxr-x 10 ec2-user ec2-user  4096 Mar  4 12:34 react_src
-rw-r--r-- 1 ec2-user ec2-user  569 Feb 15 03:57 README.md
-rw-rw-r-- 1 ec2-user ec2-user    0 Mar  4 13:07 result.txt
drwxrwxr-x 2 ec2-user ec2-user    6 Mar  4 12:36 why
```

출력되는 방향을 다른 곳으로 돌려서 파일을 저장시켰다. 이를 redirection 이라고 한다.

Review: UNIX Programs

- **Means of input :**

- Program arguments
[control information]
- Environment variables
[state information]
- Standard input [data]



- **Means of output :**

- Return status code [control information]
- Standard out [data]
- Standard error [error messages]

ls : program

-al : commandline arguments

the result of program : standard output

redirection : program > the name of file

standard error : print error messages

```
ec2-user:~/environment $ mkdir rename2
ec2-user:~/environment $ rm -r rename2
ec2-user:~/environment $ rm -r rename2
rm: cannot remove 'rename2': No such file or directory
```

- error 는 redirection 이 적용되지 않는다.
 - 이유? > or '1>'는 standard output, 표준 출력을 의미한다.
 - 해결방법 : error 를 저장하고 싶다면 '2>'를 사용해서 monitor 가 아닌 새로운 파일로 redirection 할 수 있다.

```
ec2-user:~/environment $ rm rename2.txt 2> error.log
ec2-user:~/environment $ cat error.log
rm: cannot remove 'rename2.txt': No such file or directory
```


즉, 1 개의 input 과 2 개의 output 이 존재하는 것이다.

Input

cat 은 사용자의 입력을 받고 출력한다.

hello.txt 라는 파일이 존재할 때, input 을 다르게 받을 수도 있다.

- cat hello.txt 는 commandline argument 를 통해 출력한다.
- cat < hello.txt 는 standard input 을 통해 출력한다.

head 는 파일의 내용을 10 줄만 출력하는 기능이다.

head -n1 은 1 줄만 출력한다. -n1 : commandline argument

```
ec2-user:~/environment $ head linux.txt
리눅스(Linux)[4]는 1991년 9월 17일 리누스 토르발스가 처음 출시한 운영 체제 커널인[5][6][7] 리눅스 커널에 기반을 둔 오픈 소스 유닉스 계열 운영 체제 계열이다.[8] 리눅스는 일반적으로 리눅스 배포판 안에 패키지 처리된다.

배포판에는 리눅스 커널과 지원 시스템 소프트웨어, 그리고 라이브러리를 포함하고 있으며 이를 가운데 다수가 GNU 프로젝트에 의해 제공된다. 수많은 리눅스 배포판은 배포판 이름에 "리눅스"라는 단어를 사용하지만 자유 소프트웨어 재단은 GNU 소프트웨어의 중요성을 강조하기 위해 (이름에 일부 논란이 있기는 하지만) GNU/리눅스(GNU/Linux)라는 이름을 사용한다.[9][10]

저명한 리눅스 배포판[11][12][13]에는 데비안, 페도라, 우분투가 포함된다. 상용 배포판에는 레드햇 엔터프라이즈 리눅스(RHEL), 수세 리눅스 엔터프라이즈 서버가 포함된다. 데스크톱 리눅스 배포판에는 x11이나 웨이랜드 등의 윈도우 시스템, 그리고 그놈, KDE 플라스마 등의 데스크톱 환경이 포함된다. 서버용 배포판은 그래픽스 기능이 제외되거나 LAMP 등의 솔루션 스택이 포함될 수 있다. 리눅스는 자유로이 재배포가 가능하므로 누구든지 어떠한 목적으로든 배포판을 만들 수 있다.[14]

리눅스는 처음에는 인텔 x86 아키텍처에 기반한 개인용 컴퓨터용으로 개발되었으나 그 뒤로 기타 운영 체제 외에 더 많은 플랫폼으로 이식되었다.[15] 스마트폰에 안드로이드(리눅스 기반임)가 우세해지면서 리눅스는 또한 모든 범용 운영 체제 가운데 가장 많은 설치 기반을 보유하고 있기도 하다.[16][17] 데스크톱 컴퓨터 중 대략 2.3%만이 리눅스를 사용하지만,[18][19] 리눅스 커널 기반 크롬 OS를 구동하는 크롬북은 미국 K-12 교육 시장을 지배하며 미국 내 300달러 미만 노트북 컴퓨터 가운데 20% 가까이 차지하고 있다.[20] 리눅스는 서버 시장에서 주도적인 운영 체제이며(상위 100만 대 웹 서버 운영 체제 중 96.4% 이상이 리눅스이다),[21] 메인프레임 컴퓨터 등의 기타 빅 마인 시스템들 주도하며 TOP500 슈퍼컴퓨터에 쓰이는 유일한 운영 체제이다.(2017년 11월 이후로 모든 경쟁자를 점차적으로 제거하였다).[22][23][24]

리눅스는 또한 일반적으로 운영 체제가 펌웨어로 빌드되면서 시스템에 높은 수준으로 맞추어진 임베디드 시스템에서도 동작한다. 여기에는 라우터, 자동화 컨트롤, 스마트 홈 기술(예: 구글 네스트),[25] 텔레비전(삼성전자, LG전자의 스마트TV는 각각 타이젠과 웹OS를 사용함),[26][27][28] 자동차(예: 테슬라, 아우디, 메르세데스-벤츠, 현대자동차, 도요타는 모두 리눅스에 의존함),[29] 디지털 비디오 레코더, 비디오 게임기, 스마트워치가 포함된다.[30] 펠컨 9와 드래곤 2의 항공 전자장치는 커스터마이징된 리눅스 버전을 사용한다.[31]
```

```
ec2-user:~/environment $ head -n1 linux.txt
리눅스(Linux)[4]는 1991년 9월 17일 리누스 토르발스가 처음 출시한 운영 체제 커널인[5][6][7] 리눅스 커널에 기반을 둔 오픈 소스 유닉스 계열 운영 체제 계열이다.[8] 리눅스는 일반적으로 리눅스 배포판 안에 패키지 처리된다.
```

표준 입력과 표준 출력을 다 받는 경우

head -n1 < linux.txt > one.txt

-> linux.txt 의 1 번째 줄을 standard input 으로 출력 후 one.txt 에 standard output 으로 저장함.

```
ec2-user:~/environment $ head -n1 < linux.txt > one.txt
ec2-user:~/environment $ cat one.txt
리눅스(Linux)[4]는 1991년 9월 17일 리누스 토르발스가 처음 출시한 운영 체제 커널인[5][6][7] 리눅스 커널에 기반을 둔 오픈 소스 유닉스 계열 운영 체제 계열이다.[8] 리눅스는 일반적으로 리눅스 배포판 안에 패키지 처리된다.
```

append

'>>' redirection 의 결과를 덧댄다라고 생각하면 된다.

'<<' 여러개의 입력을 합치는 것이다.

'>' /dev/null 로 사용하면 이는 그냥 출력을 버린다고 생각한다.