

SWE3004 Operating Systems, Fall 2025

# Project 0. Introduction of xv6

---

TA)

Gwanjong Park

Yunseong Shin

# Project plan

## Schedule and points

	Assignment	Schedule	Point
0	Booting xv6 operating system	9/10 ~ 9/23	10
1	System call	9/17 ~ 9/30	15
2	CPU scheduling	10/1 ~ 10/19	25
3	Virtual memory	10/29 ~ 11/11	25
4	Page replacement	11/12 ~ 11/25	25
5	File systems	11/26 ~ 12/9	0 (tentative)

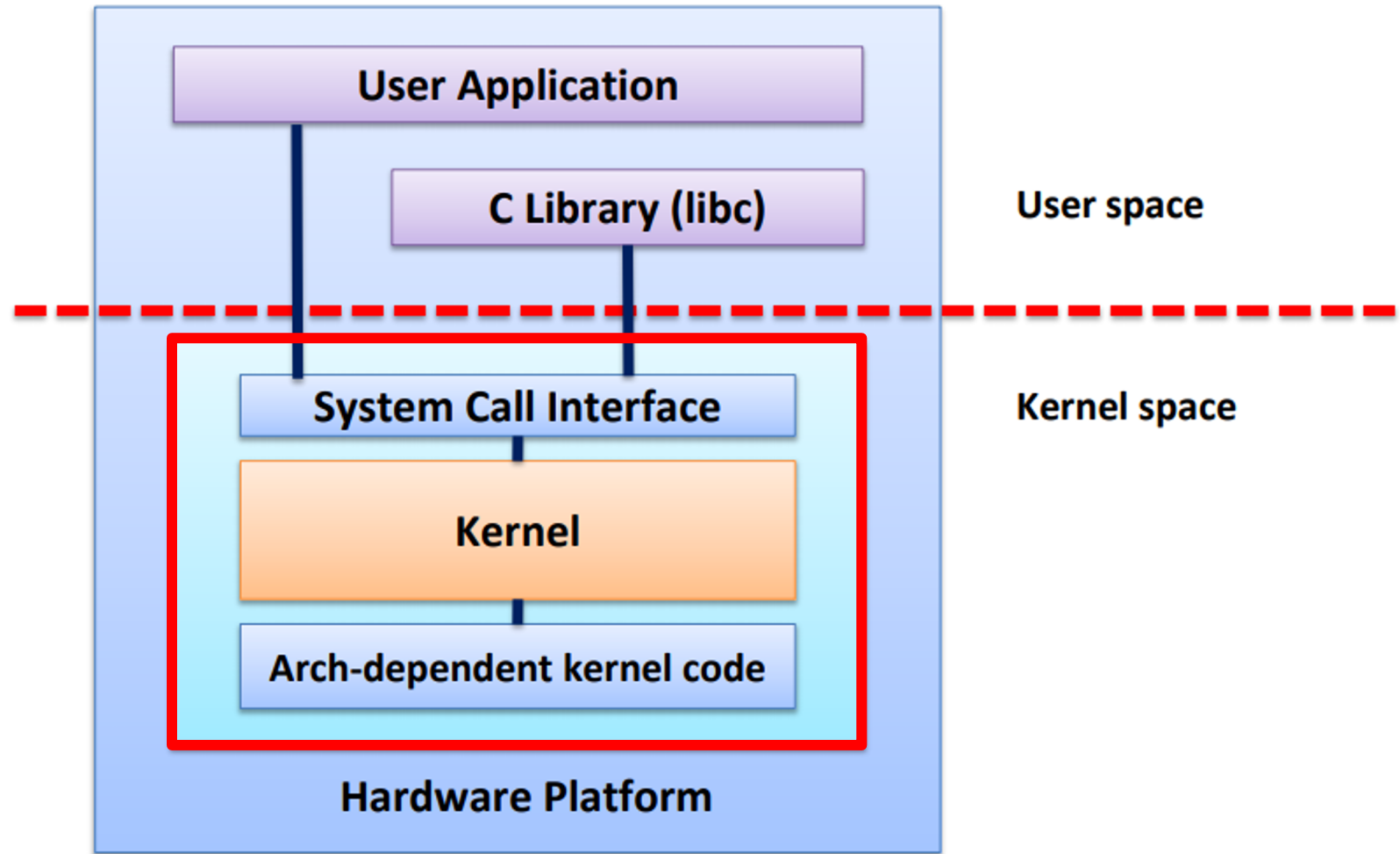
# Project plan

Total 6 projects

0) Booting xv6 operating system

- 1) System call
- 2) CPU scheduling
- 3) Virtual memory
- 4) Page replacement
- 5) File systems

# Operating System



# xv6 Operating System

Unix-like teaching operating system developed by MIT

Designed for the RISC-V architecture

```
xv6 kernel is booting
hart 2 starting
hart 1 starting
init: starting sh
$ ls
.          1 1 1024
..         1 1 1024
README    2 2 2292
cat        2 3 35544
echo      2 4 34360
forktest  2 5 16384
grep       2 6 38992
init       2 7 34840
kill       2 8 34304
ln         2 9 34112
ls         2 10 37640
mkdir     2 11 34360
rm         2 12 34352
sh         2 13 57136
stressfs  2 14 35232
usertests  2 15 181720
grind      2 16 50608
wc         2 17 36448
zombie    2 18 33704
console   3 19 0
$
```

*xv6 shell status with ls command*

# Development Environment

- Ye Server
  - Problems caused by development in other environments will be your own risk.
- Connection method example
  - If your student id is 2024123456:  
`$ ssh s24123456@swye.skku.edu -p 1398`
  - If you are new to Ye server, your password is initially set to  
`pw24123456`

# Clone & booting xv6

- Clone the official xv6 RISC-V repository
  - `$ git clone https://github.com/mit-pdos/xv6-riscv.git`
- Run xv6
  - `$ cd xv6-riscv`
  - `$ make qemu`
- Terminate xv6
  - 'Ctrl + A', then press 'X'

# Project 0. Booting xv6

Print your student id, name, and any message (optional) on boot message

```
xv6 kernel is booting  
  
hart 2 starting  
hart 1 starting  
init: starting sh  
Student ID: 2024123456  
Name: Gildong Hong  
=====Your message=====  
$
```

*xv6 booting message example*

Hint : Use grep or other analysis tool



# Submission

- Use the *submit* file on Ye Server

Make sure to clean your build before submitting:

- \$ make clean
- \$ ~swe3004/bin/submit pa0 xv6-riscv
- If you renamed the folder, replace 'xv6-riscv' with the new folder name.

```
s25315691@swye:~/os2025$ ~swe3004/bin/submit pa0 xv6-riscv
user name           :s25315691
Submitted Files for pa0:
File Name                                File Size      Time
-----
pa0-s25315691-Mar.11.17.10.611413768    74312          Tue Mar  1 17:10:00 2025
```

*successful submission example*

# Check Submission

- Use the *check-submission* file on Ye Server

You can submit several times, and the submission history can be checked through check-submission

– \$ ~swe3004/bin/check-submission pa0

- Only the last submission will be graded

```
s25315691@swye:~/os2025$ ~swe3004/bin/check-submission pa0
Submitted Files for pa0:
File Name                                File Size    Time
-----
pa0-s25315691-Mar.11.17.10.611413768    74312        Tue Mar  1 17:10:00 2025

Your last submission
-----
[#]          File Name                                File Size
-----
```

*submission check example*

# Submission Errors

- Please check that the submission was successful. Note that file sizes may vary slightly and do not have to be exactly the same.

```
Submitted Files for pa0:
```

File Name		File Size	Time
pa0-s25315691-Mar.11.17.40.423220757	①	45	Tue Mar 1 17:40:41 2025
pa0-s25315691-Mar.11.17.39.496785533	②	122	Tue Mar 1 17:39:12 2025
pa0-s25315691-Mar.11.17.10.611413768	③	74312	Tue Mar 1 17:10:00 2025

*submission examples*

- ① Submit from an incorrect directory
- ② Submitted an empty file
- ③ Submitted xv6 successfully

# Submission

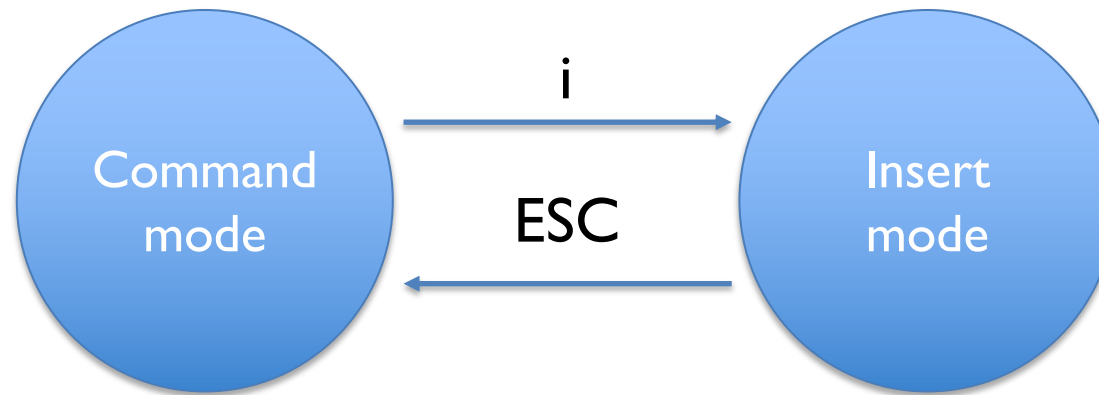
- PLEASE DO NOT COPY
  - We will run **inspection program** on all the submissions
  - Any unannounced penalty can be given to **both students**
    - 0 points / negative points / F grade ...
- Due date: **9/19(Fri.)**, 23:59:59 PM
  - Late Submission Penalty: **-25%** per day

# Questions

- If you have questions, please ask on iCampus
  - Please use the **discussion board** instead of direct messages
- You can also visit Corporate Collaboration Center (#85533)
  - Please send an iCampus message to the TA before visiting
- Reading the xv6 commentary will be very useful:  
<https://pdos.csail.mit.edu/6.828/2023/xv6/book-riscv-rev3.pdf>

# Appendix. Vim

- Vi Improved
- Text Editor program for Unix
  - `sudo apt-get install vim`



# Useful Vim command

- In command mode

Command	Operation
<code>:w</code>	save changes to a file
<code>:q</code>	quit Vim
<code>:wq</code>	save changes and quit Vim
<code>/[string]</code>	search the string from forward
<code>?[string]</code>	search the string from backward
<code>:sp</code>	Divide the window (horizontal)
<code>:vs</code>	Divide the window (vertical)
<code>ctrl + w -&gt; w</code>	move window-to-window

Command	Operation
<code>dd</code>	delete the line
<code>yy</code>	copy the line
<code>P</code>	paste

# Vim Setting

- Vim Preference Setting
  - vimrc : settings file used by Vim

Command	Operation
set nu	display line number
set nonu	hide line number
set mouse=a	enable auto visual mode using mouse
set mouse-=a	disable auto visual mode using mouse
set autoindent	set automatic indentation
set ts=[#]	adjust tap size
set et	expand tap

~/.vimrc

```
set et ts=4
set nu
set mouse=a
set autoindent
```



# Source code analysis tool

- Ctags, Cscope are useful when analyzing source code
  - vim editor supports ctags and cscope
- **Ctags**
  - Recognizes **symbols** in source code and links related files
  - Allows quick and easy **navigation** to function and variable definitions
  - Helps **text editors** (such as Vim) locate functions efficiently
- **Cscope**
  - Searches for **functions, function calls, and strings**
  - Finds **where functions are defined and used**
  - Efficiently explores **large codebases**

# Ctags

- Vim setting for ctags
  - \$ sudo apt-get install ctags
  - \$ ctags -R (use in top-level kernel source directory)
  - \$ vi ~/.vimrc
  - Add “set tags=[Location of tag file]/tags”
    - Ex)

```
set tags=/home/seungwoo/xv6-sse-local/tags
```



check your path!

# Ctags

- Ctags usage (in Vim)
  - Ctrl + ] : follow tag

```
for(i=1; i<argc; i++)  
    kill(atoi(argv[i]));  
exit();
```

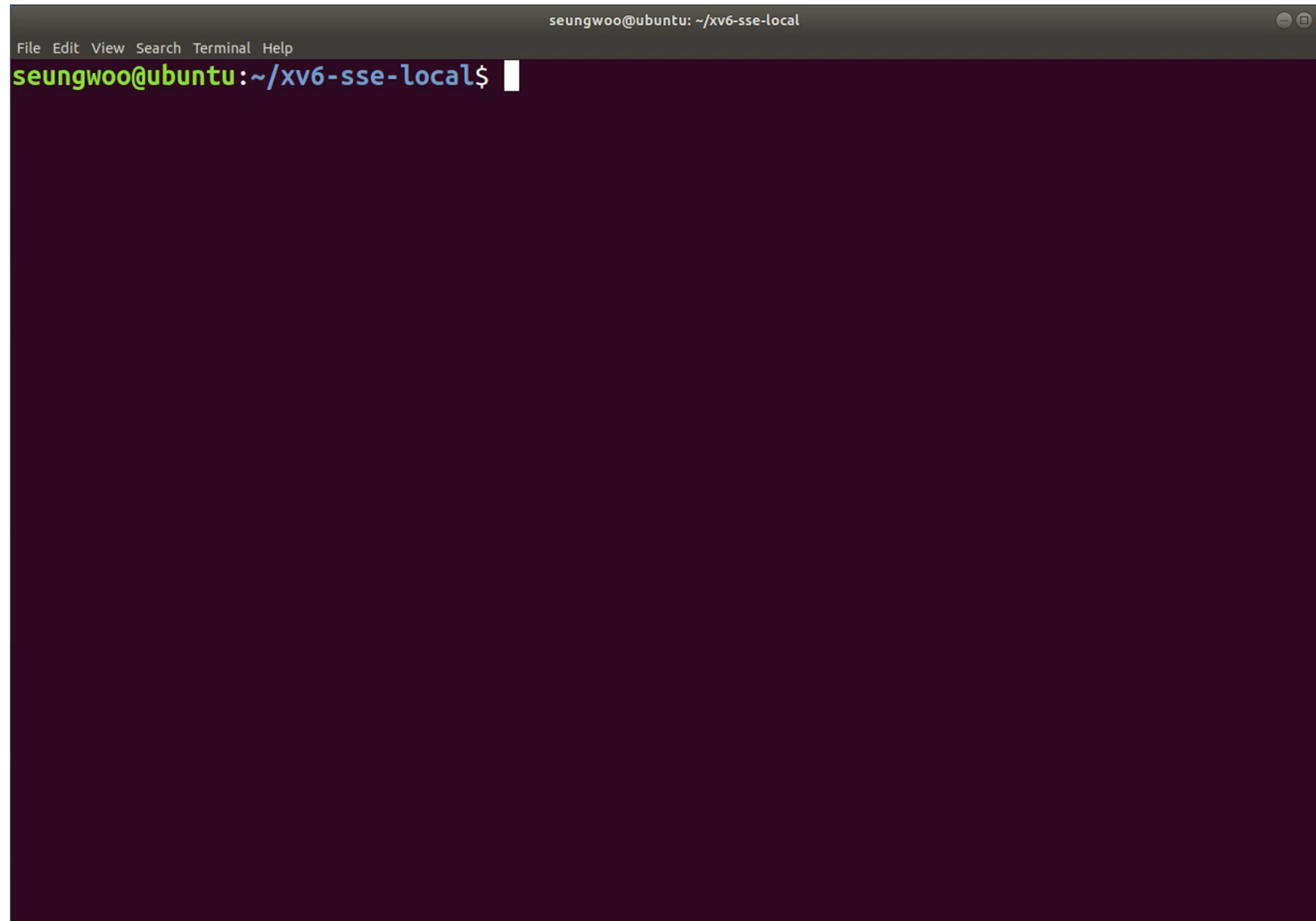
```
int  
kill(int pid)  
{  
    struct proc *p;  
    acquire(&ptable.lock);
```

- Ctrl + T : back to last tag

```
int  
kill(int pid)  
{  
    struct proc *p;  
    acquire(&ptable.lock);
```

```
for(i=1; i<argc; i++)  
    kill(atoi(argv[i]));  
exit();
```

# Ctags

A terminal window titled "seungwoo@ubuntu: ~/xv6-sse-local" with a menu bar (File, Edit, View, Search, Terminal, Help). The prompt "seungwoo@ubuntu:~/xv6-sse-local\$" is displayed on a dark purple background.

```
seungwoo@ubuntu: ~/xv6-sse-local$
```

# Cscope

- Vim setting for cscope
  - \$ sudo apt-get install cscope
  - \$ cscope -R (use in top-level kernel source directory)
  - Ctrl + D
  - vi ~/.vimrc
  - Add

```
set cst  
cs add /home/seungwoo/xv6-sse-local/cscope.out /home/seungwoo/xv6-sse-local/
```



Check your path!

# Cscope

- Cscope usage (in Vim)
  - :cs find [type] [search]

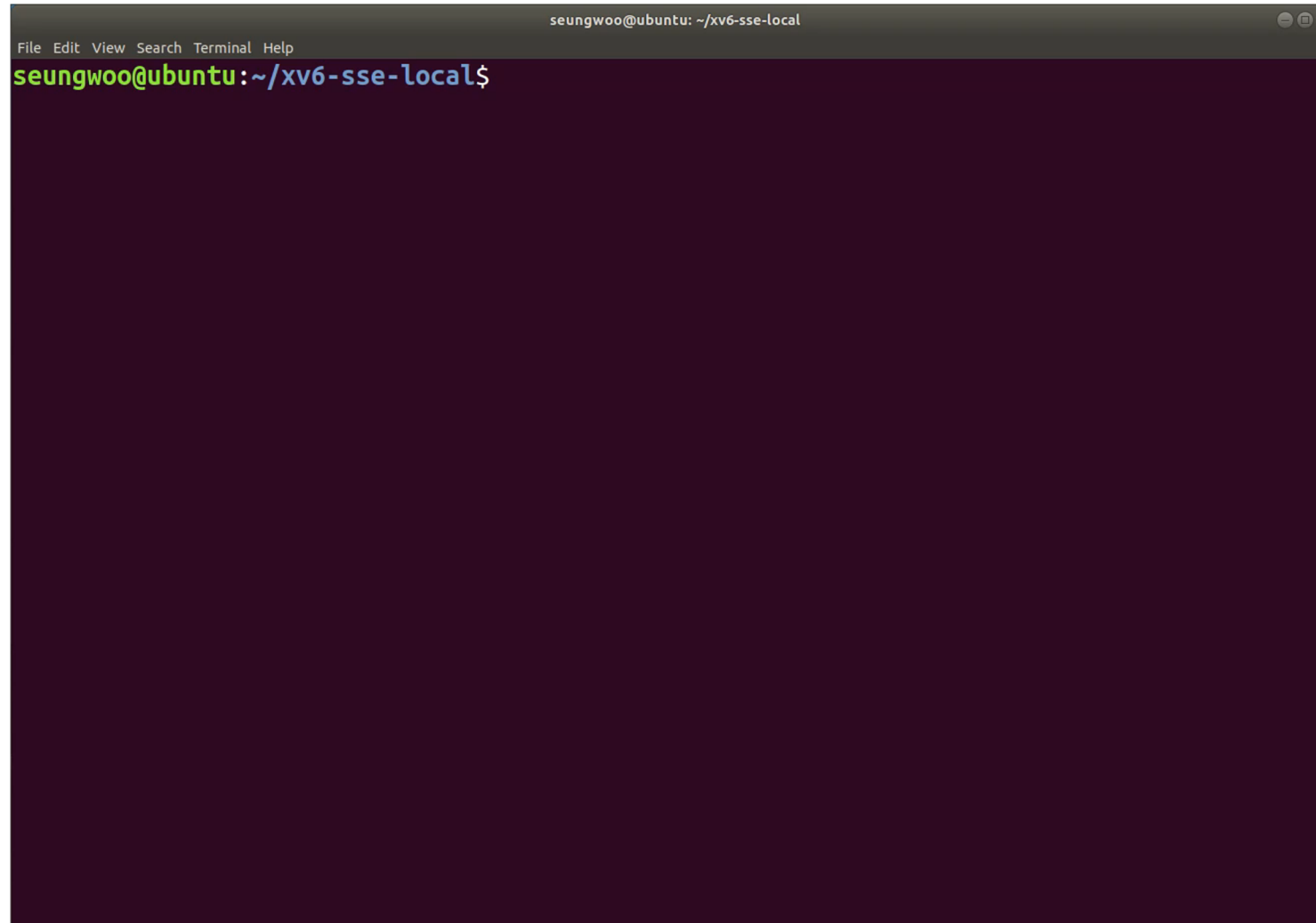
type	Operation
0 or s	Find this C symbol
1 or g	Find this definition
2 or d	Find functions called by this function
3 or c	Find functions calling this function
4 or t	Find assignments to

- ex) :cs find c [search function]

```
:cs find c fetchint
```

```
Cscope tag: fetchint
#  line  filename / context / line
1   155  defs.h <<panic>>
      int fetchint(uint, int *);
2   155  defs.h <<scheduler>>
      int fetchint(uint, int *);
```

# Cscope



# Any other useful shell command


- Grep usage

- `grep -nR “[string to search]”`

- Find the file where the string exists and also show line number
    - Easier to use than cscope

- Ex)

```
grep -nR fetchint
```



```
syscall.c:18:fetchint(uint addr, int *ip)  
syscall.c:52: return fetchint((myproc()->tf->e  
kernel.sym:161:801047e0 fetchint  
dist/defs.h:154:int fetchint(uint,  
dist/syscall.c:18:fetchint(uint addr, int *ip)
```

- Silversearcher-ag usage

- `$ ag [string to search]`