Project 1: NPShell

NP TA 政毅

Important Dates

- **Deadline: 10/21 (Fri.) 23:55**
- **Demo: 10/22 (Sat.) at EC 315**
  - Test your program and ask some questions (80%)
  - Implement 1 or 2 extra functions with limited time (20%)
- **Office hour: 10/11 & 10/18 (Tue.) 13:00 - 15:00**
  - Online discussion using google meet
  - You should send an email to all the TAs to make an appointment

About Questions

- You are **HIGHLY** encouraged to ask your questions on E3 **Project 1 channel**.
  - Check the spec and other questions first
- For personal problems, you can mail to **all** the TAs:
  - franklp97531@gmail.com
  - yoway.cs10@nycu.edu.tw
  - sciencethebird@gmail.com
  - pikachin.cs10@nycu.edu.tw

Reminders

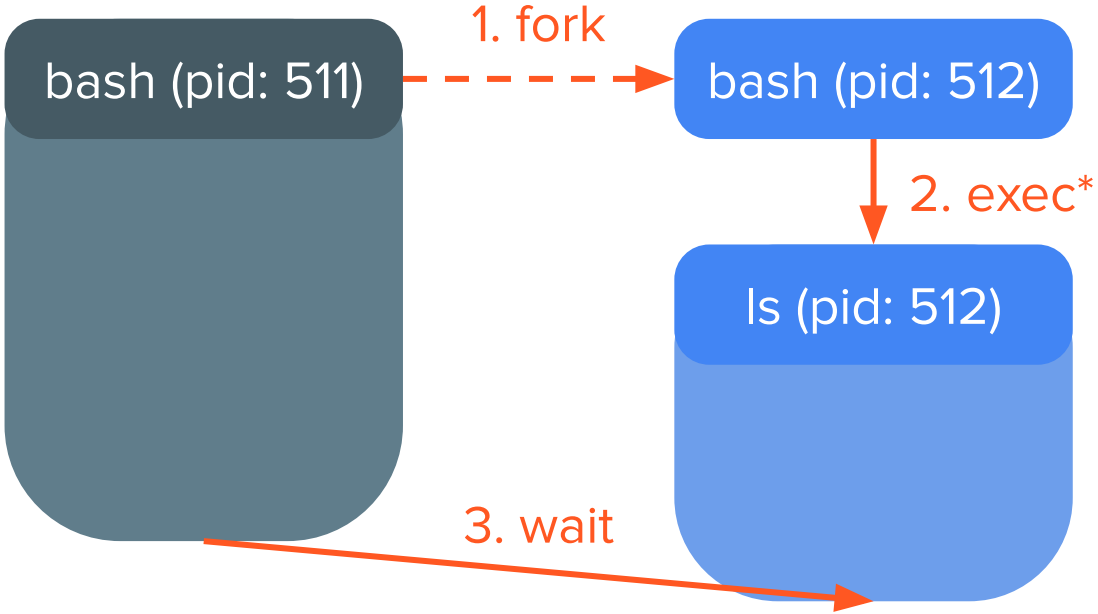
- Make sure you are invited to our Bitbucket workspace
- Make sure you have an account for NYCU CSIT workstation
  - You need this account to login to NP servers

NP Servers

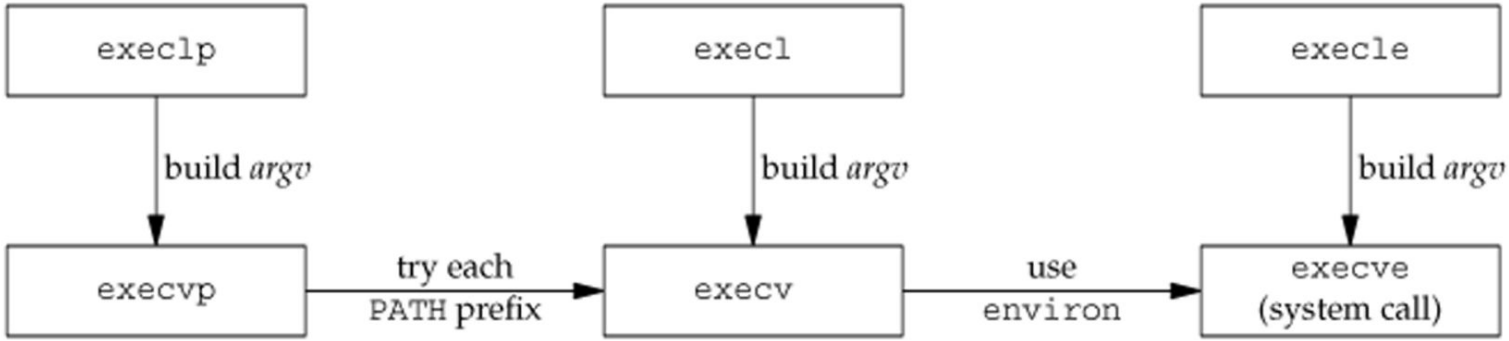
- **Your projects will be run on NP servers during demo**
- NP projects should run on NP servers
- Any abuse of NP servers will be recorded
- Do not leave any zombie process in the system

Implementation

Spawn a New Process

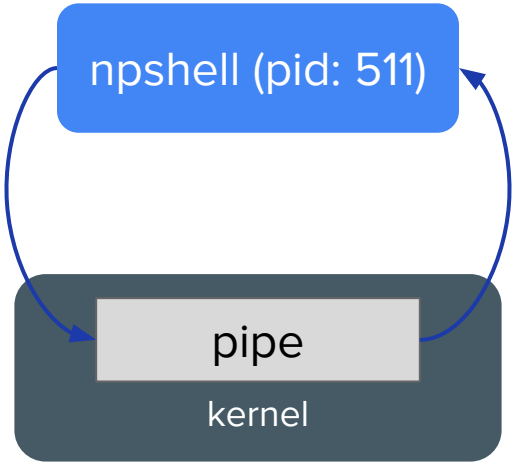


The exec family of functions

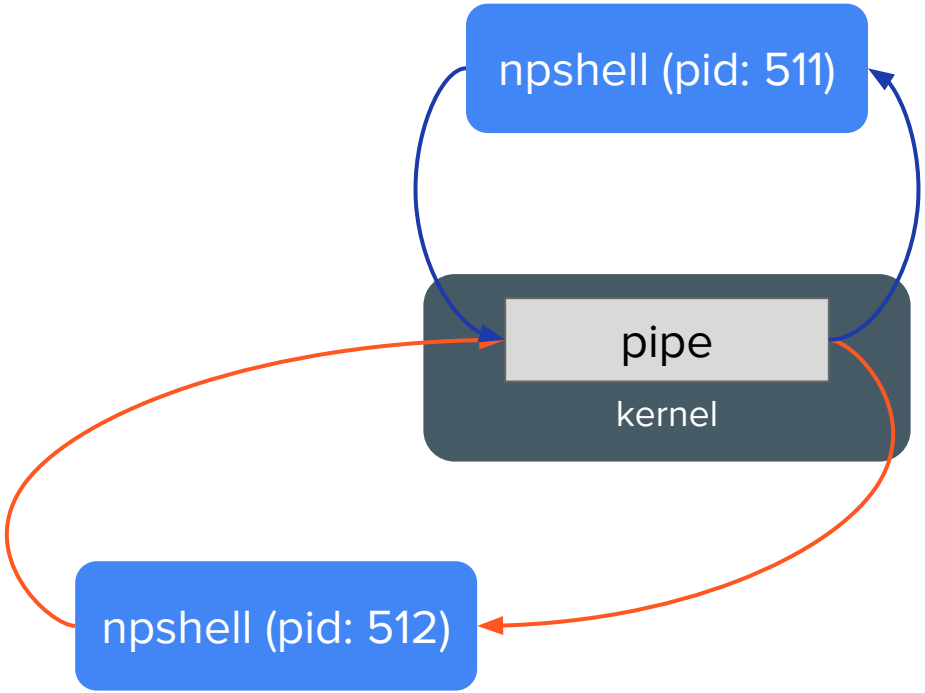


Pipe Between Processes

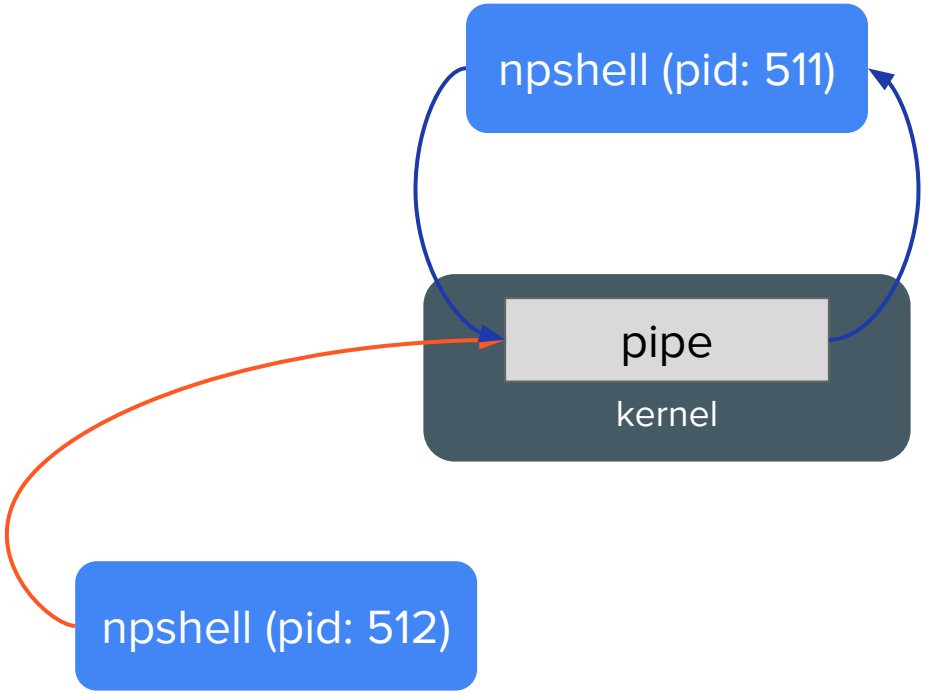
Pipe Between Processes - pipe



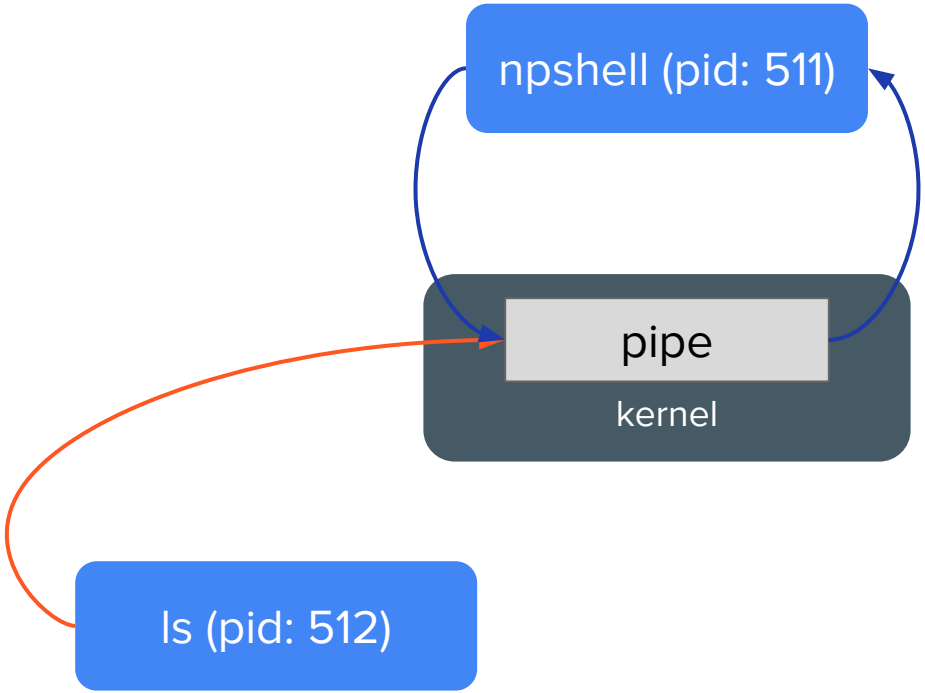
Pipe Between Processes - fork



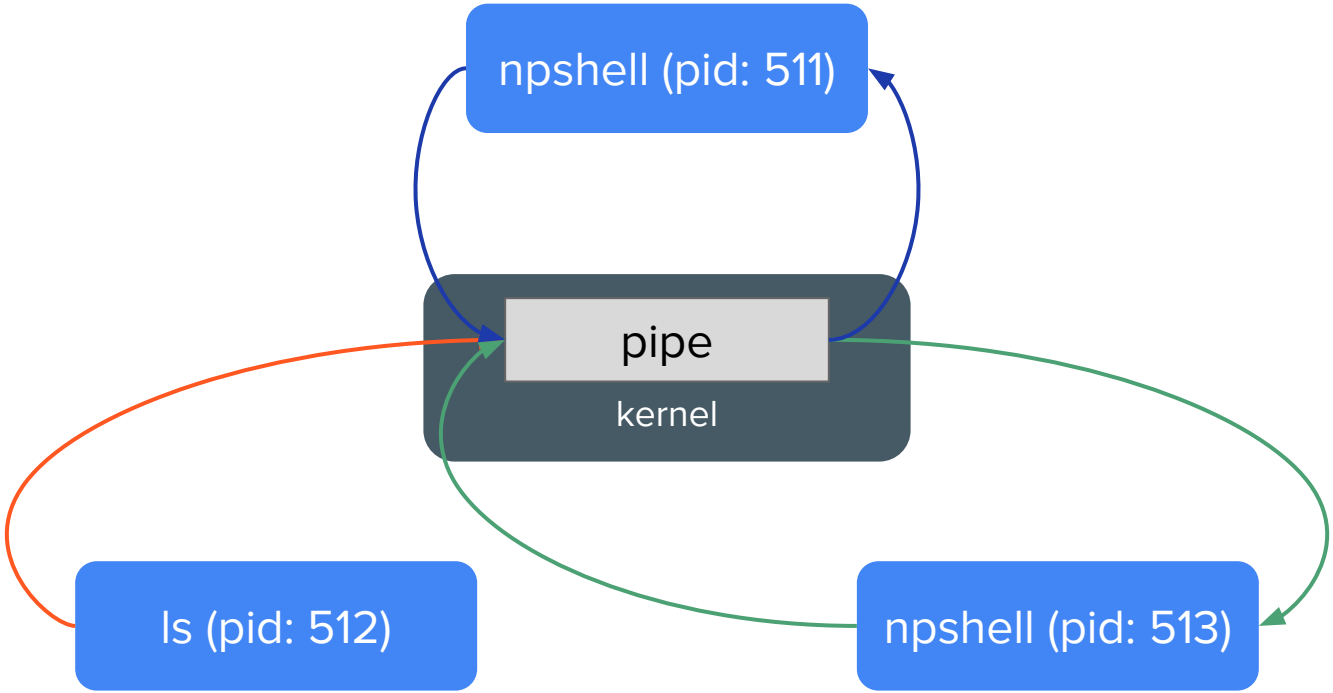
Pipe Between Processes - close



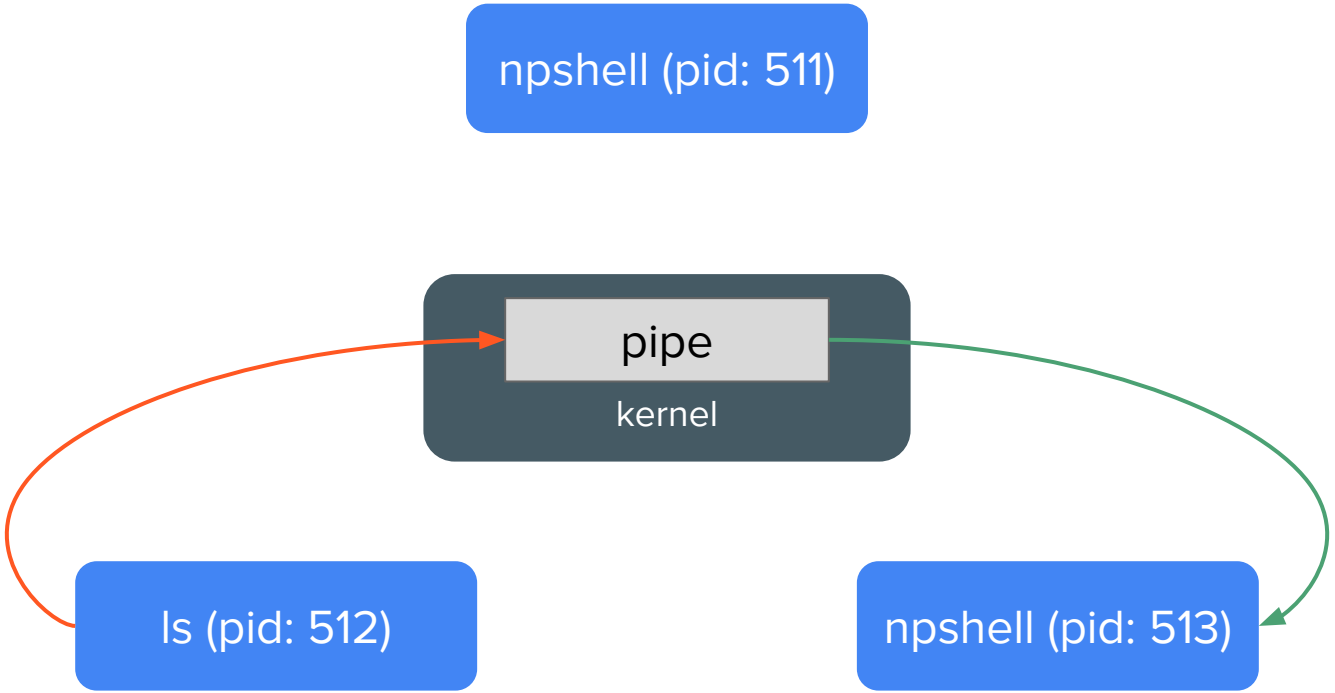
Pipe Between Processes - exec



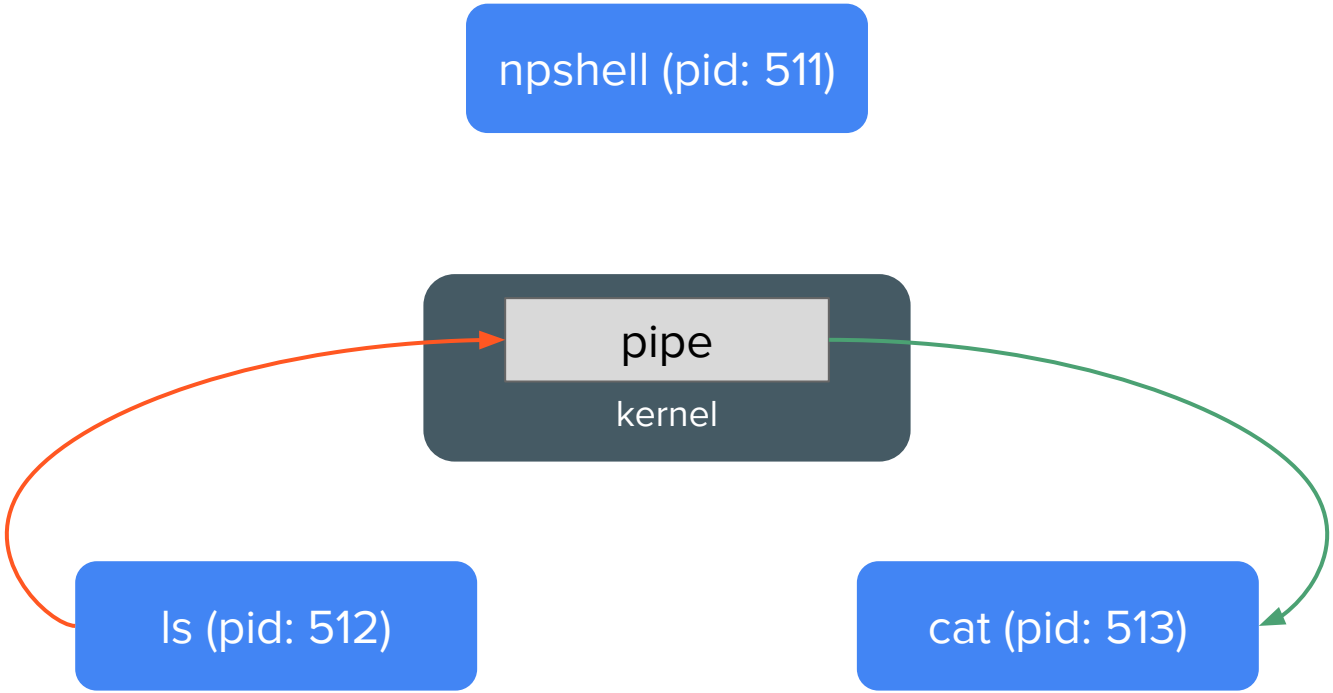
Pipe Between Processes - fork



Pipe Between Processes - close

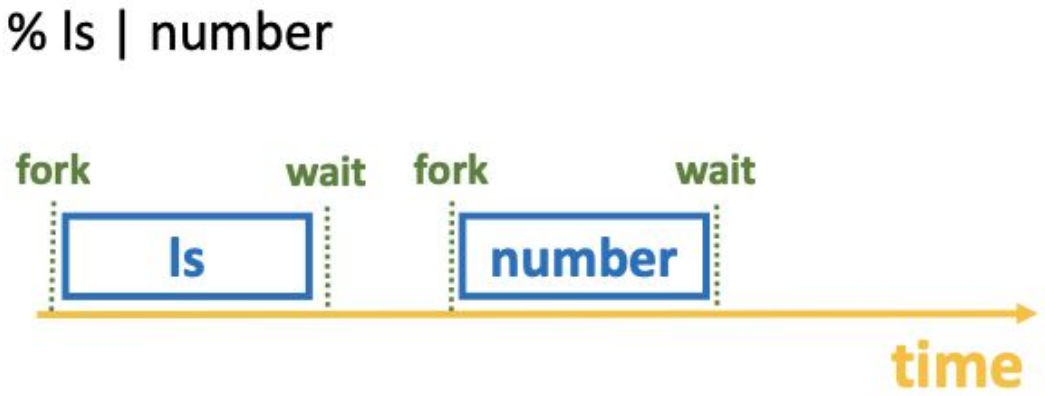


Pipe Between Processes - exec



Issues

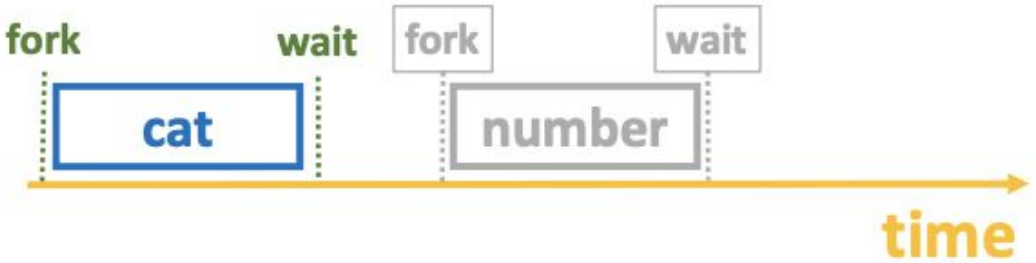
Implementation: Wait for each child process



Problem: Unable to process large data

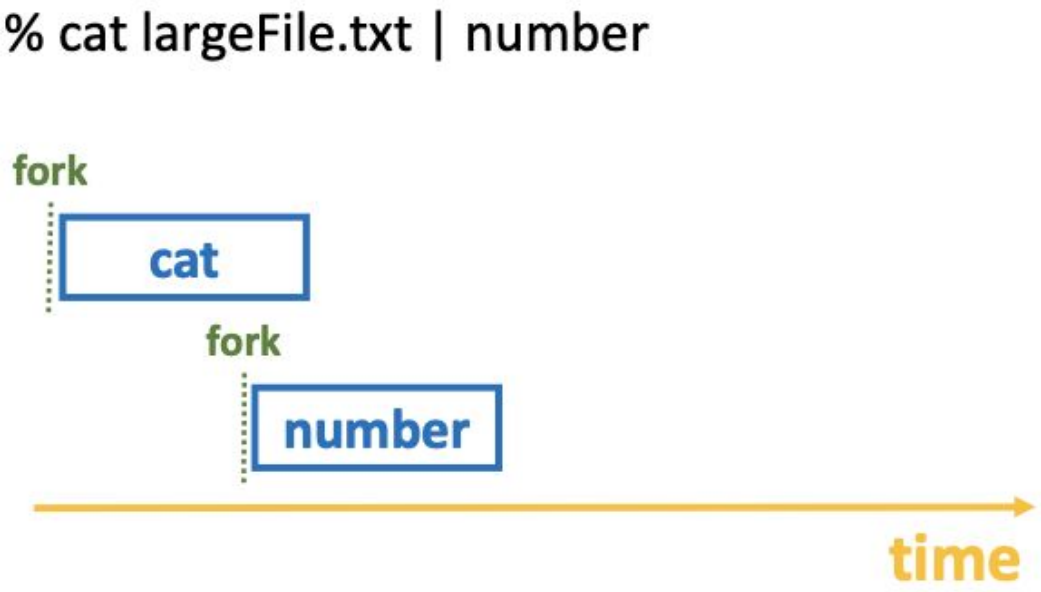
```
% cat largeFile.txt | number
```

```
% cat largeFile.txt |1
% number
```

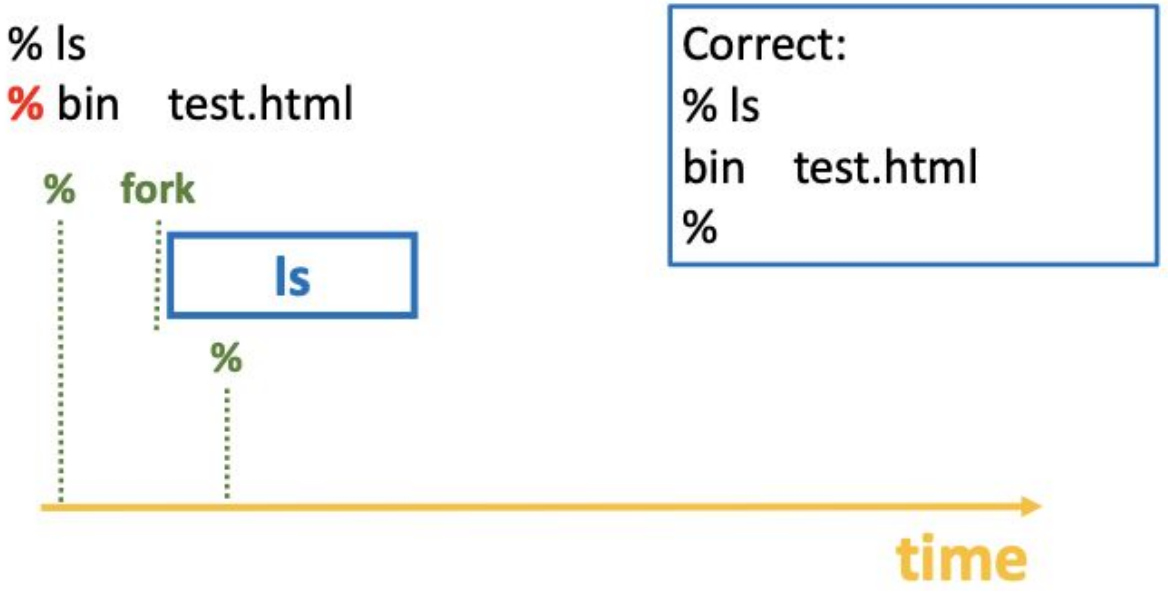


The process will hang forever !

Implementation: Never wait for child processes

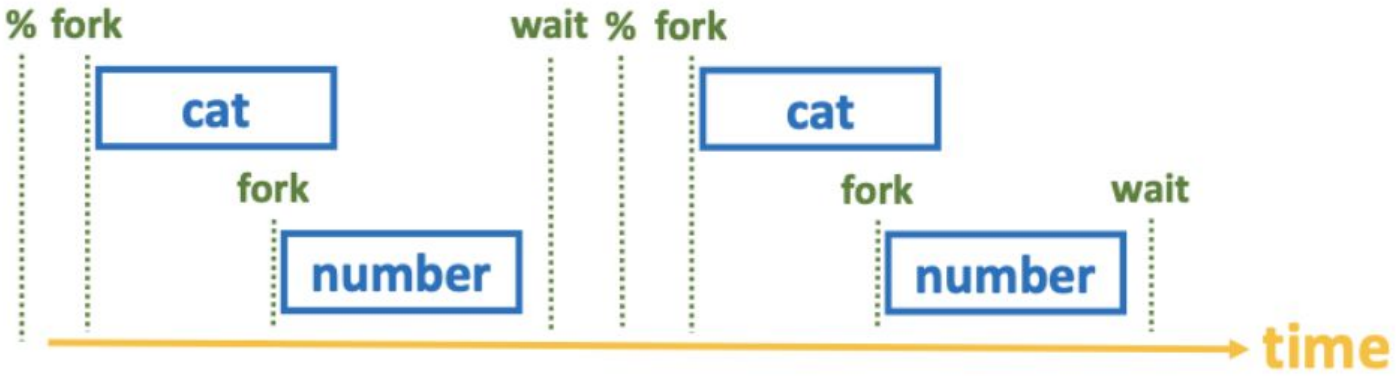


Problem: Prompt ordering and zombie processes



Implementation: Wait for all piped commands together

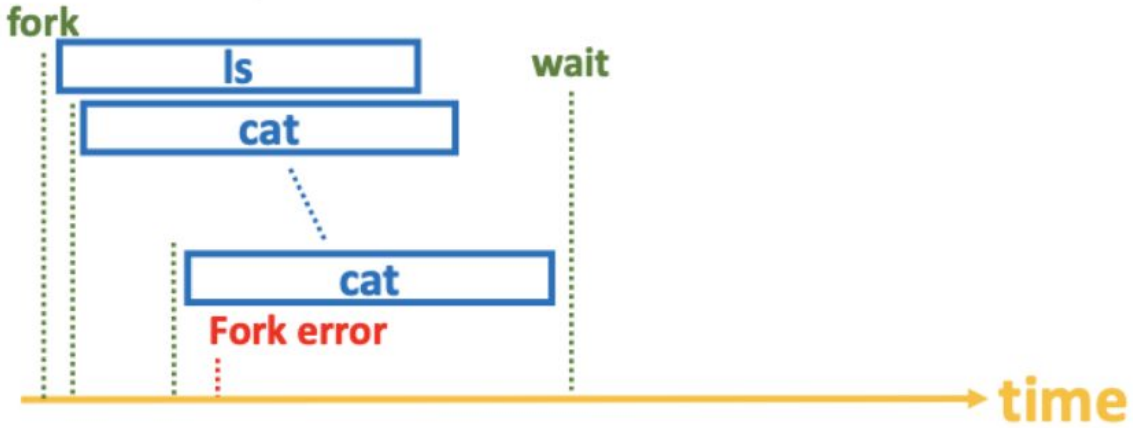
```
% cat largeFile.txt | number
% cat largeFile.txt |1
% number
```



Problem: Process limitation

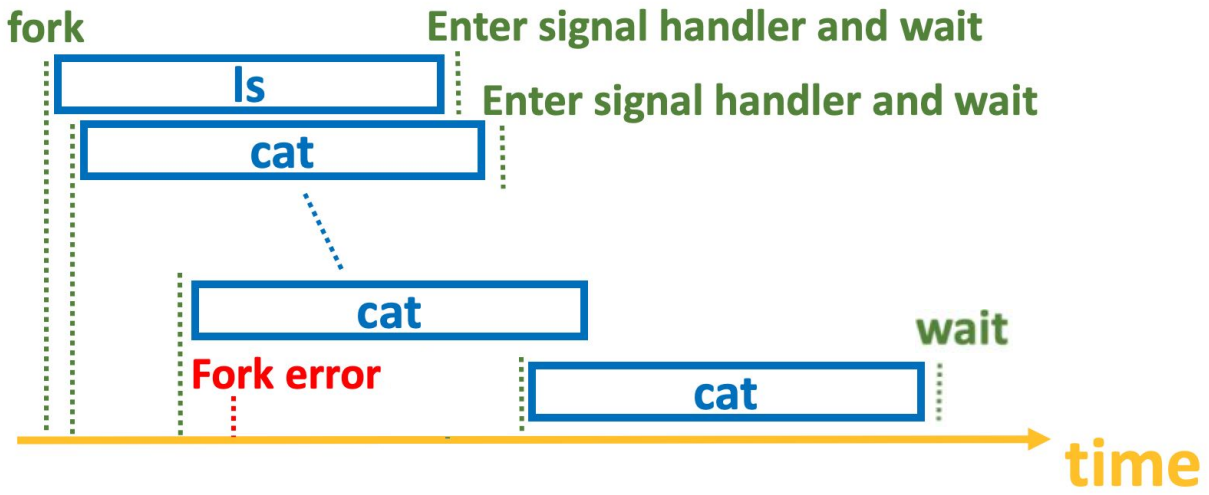
Suppose the process limit is **512** and the number of piped commands is **1000**...

ls | cat | cat | cat | cat ..... cat | cat



One Possible Implementation: Use signal handler

ls | cat | cat | cat | cat ..... cat | cat



Hints

- Think about when the npshell should wait the child process
- Try to understand the concept of parent/child processes, zombie processes, SIGCHLD, and signal handler
- Failure (e.g., fork error) needs to be handled

Hints

- Functions you may use
  - fork
  - pipe
  - dup, dup2
  - close
  - exec*
  - wait, waitpid
  - signal
  - setenv
  - getenv