

# Systems Programming

Programming HW3

連彥傑  
侯廷璋  
林家瑞  
江建德

# Task

- Handle signals with priority
- Implement receiver.c
- Implement sender.c (option)

# Problem Description

- Document transferring system
- Three type of documents:
  - ordinary document
  - urgent document
  - very urgent document
- The priority of document is:
  - very urgent > urgent > ordinary
- A document with higher priority should be dealt with first when the receiver receives it.

# Problem Description

- Processing time of document:
  - ordinary document: 1.0 second
  - urgent document: 0.5 second
  - very urgent document: 0.2 second
- Finishing time of document:
  - ordinary document: no limit
  - urgent document: 1.0 second
  - very urgent document: 0.3 second

# receiver.c

- **\$/receiver [test\_data]**
  - test\_data is a file name . It is also the argument of the sender program .
- fork() one child and run exec() to execute the sender program
- open a file named with “receiver\_log” to record the action of receiver
  - start dealing with the document:
    - write “receive [priority] [serial\_number]”
  - receiver finishes the document :
    - write “finish [priority] [serial\_number]”

# receiver.c

- When receiver read “EOF” from pipe
  - it should write “terminate” to file “receiver\_log”
  - terminate the program itself.
- When receiver receives the signal “SIGINT” from keyboard:
  - it should kill sender and write “terminate” to file “receiver\_log”
  - terminate the program itself.

# sender.c

- **\$/sender [test\_data]**
- read **[test\_data]** and send the document at the specific time in the **[test\_data]**
- open a file named with “sender\_log” to record the action of sender
  - sends the document:
    - write “send [priority] [serial\_number]”
  - gets the finished document:
    - write “get [priority] [serial\_number]”
- If receiver don't send back the document within the specific time:
  - the sender should write “timeout [priority] [serial\_number]” to file “sender\_log”
  - terminate the program itself.

# From receiver to sender

- The “SIGINT” signal indicates an ordinary document is finished.
- The “SIGUSR1” signal indicates an urgent document is finished.
- The “SIGUSR2” signal indicates a very urgent document is finished.



# From sender to receiver

- Send “ordinary\n” through pipe to indicate an ordinary document sent.
- The “SIGUSR1” signal indicates an urgent document sent.
- The “SIGUSR2” signal indicates a very urgent document sent.

# Format of test\_data

- Each line has two numbers
  - the first number is the priority of document (0: ordinary, 1: urgent, 2: very urgent)
  - the second number is the sending time of the document which is in ascending order.  
(unit is 0.1 second)

# Format of receiver\_log

- **[action] [priority] [serial\_number]**
  - **[action]** will be “receive” or “finish”
  - **[priority]** will be 0, 1, or 2
  - **[serial\_number]** is the serial number of document

# Format of sender\_log

- **[action] [priority] [serial\_number]**
  - **[action]** will be “send” or “finish” or “timeout”
  - **[priority]** will be 0, 1, or 2
  - **[serial\_number]** is the serial number of document

# Sample execution

- **test\_data:**

- 0 0

- 1 8

- 2 12

- 2 21

- 1 22

# Sample execution

- **receiver\_log:**

- receive 0 1
- receive 1 1
- receive 2 1
- finish 2 1
- finish 1 1
- finish 0 1
- receive 2 2
- finish 2 2
- receive 1 2
- finish 1 2
- terminate

# Sample execution

- **sender\_log:**

- send 0 1
- send 1 1
- send 2 1
- finish 2 1
- finish 1 1
- finish 0 1
- send 2 2
- send 1 2
- finish 2 2
- finish 1 2

# Tasks and scoring

- Generate the two execution files. (1 point)
- Simple test data (only ordinary documents) (1 point)
- Complicated test data (ordinary documents and urgent documents) (1 point)
- More complicated test data (three types of document) (2 points)
- Implement the sender.c (2 points)
- Deal with the “SIGINT” signal (keyboard to receiver) (1 point)



# Submission

- Submit **SPHW3\_[student id].tar.gz**  
(only .tar.gz) to Ceiba  
(ex : SPHW3\_b03902000.tar.gz)
- The file should at least three files.
  - receiver.c
  - Makefile (as well as other \*.c)
  - README.txt
  - sender.c (option (2 points in this assignment))

# Punishments

- Plagiarism punishment
- Late punishment
  - 5% for each day delay
- File error punishment
  - Submission file format is .tar.gz
  - Include at least three files