**Bài ôn tập 1:**

1. **Giả định**

* Buffer không giới hạn độ dài.
* Mỗi ngăn chứa của buffer không giới hạn kiểu dữ liệu.

1. **Ý tưởng**

* Định nghĩa cấu trúc Buffer chứa một buffer và các thông tin liên quan khác. Trong đó có thông tin độ dài của buffer (length).
* Lấy input của chương trình, kiểm tra input đó, nếu không phải là số nguyên dương thì xuất thông báo và kết thúc chương trình.
* Tạo ra một shared memory và khởi tạo một buffer rỗng.
* Tạo 2 function parent\_process và child\_process:
  + parent\_process: mỗi khi buffer có một con số mới được thêm vào, xuất ra màn hình con số đó, đồng thời kiểm tra con số đó nếu bằng 1 thì kết thúc function.
  + child\_process: Liên tục tạo ra con số kế tiếp theo thuật toán mà đề đã cho, thêm con số vào buffer, đồng thời kiểm tra con số đó nếu bằng 1 thì kết thúc function.
* Dùng lệnh fork() để tạo tiến trình con.
* Tiến trình cha sẽ gọi function parent\_process và tiến trình con sẽ gọi function child\_process.
* Cuối cùng là đóng kết nối và giải phóng bộ nhớ.

1. **Code**

**A screenshot of a computer code

Description automatically generated**

*Khai báo thư viện*

**A computer code with colorful text

Description automatically generated**

*Khai báo cấu trúc Buffer*

* Cấu trúc của **Buffer** bao gồm 1 **buffer** có độ dài **1001 phần tử** và thuộc tính **length** cho biết độ dài đã ghi vào buffer.

**A computer screen shot of a program code

Description automatically generated**

*Function parent\_process*

* Được chạy bởi tiến trình cha.
* Tham số đầu vào bao gồm 1 con trỏ **Buffer**.
* Khởi tạo một dependency là **length**, mỗi khi thuộc tính **length** của **buffer** bị thay đổi thì sẽ thực thi đoạn code xử lí.
* Bên trong dòng **while(1)**, dòng **while đầu tiên** được dùng để đợi cho đến khi thuộc tính **length** của **Buffer** bị thay đổi.
* Lấy ra phần tử tiếp theo của **buffer** mà tiến trình cha chưa xuất ra màn hình. Kiểm tra số đó nếu bằng 1 thì xuất ra màn hình như là một phần tử cuối cùng và kết thúc dòng **while(1)**, còn nếu khác 1 thì vẫn xuất ra màn hình như một phần tử bình thường.
* Kết thúc dòng while, đợi cho đến khi tiến trình con kết thúc thì mới thoát function **parent\_process**.

**A computer screen shot of a program code

Description automatically generated**

*Function child\_process*

* Được sử dụng vởi tiến trình con.
* Tham số đầu vào bao gồm 1 con trỏ **Buffer** và một **giá trị xuất phát** (do người dùng nhập vào).
* Lưu giá trị xuất phát đó như phần tử đầu tiên và lưu nó vào biến num để tính toán.
* **While(num > 1)**: kiểm tra nếu **num > 1** thì sẽ thực thi thuật toán bên trong.
* Dòng **if else** bên trong dòng while: Dùng biến **num** để thực thi thuật toán của **Collatz** và lưu kết quả vào biến **num**.
* Sau đó, lưu biến **num** vào phần tử tiếp theo của **Buffer**.

**A screen shot of a computer program

Description automatically generated**

*Hàm main: Lấy và check input*

* Input của người dùng sẽ được lưu vào **agrv[1]** dưới dạng **char\***.
* Để chuyển từ **char\*** thành **int**, sử dụng hàm atoi được cung cấp sẵn trong bộ thư viện chuẩn.
* Kiểm tra nếu **start <= 0** thì xuất ra thông báo và kết thúc chương trình.

**A computer code on a black background

Description automatically generated**

*Hàm main: Khởi tạo, cấp phát shared memory và Buffer.*

* Tạo ra một shared memory bằng shm\_open với quyền **O\_CREAT**, cấp quyền đọc ghi bằng **O\_RDWR** và truyền mã quyền là **0666**.
* Cấp phát không gian cho shared memory bằng **ftruncate** với kích thước được cấp phát là độ lớn của cấu trúc Buffer.
* Ánh xạ vùng nhớ buffer vào con trỏ ptr bằng **mmap**, với vị trí ánh xạ đầu là **0** và độ lớn ánh xạ bằng với độ lớn của cấu trúc Buffer. Cấp quyền đọc ghi bằng **PROT\_READ** và **PROT\_WRITE**, cấp quyền chia sẽ thay đổi bằng **MAP\_SHARED**.
* Khởi tạo trạng thái ban đầu cho **Buffer** với **length = 0** (buffer chưa có phần tử nào được thêm vào).

**A computer code with colorful text

Description automatically generated**

*Hàm main: Tạo tiến trình con và chia công việc*

* Lệnh **fork** để tạo tiến trình còn.
* Với **pid == 0**, đây là ngữ cảnh của tiến trình con. Thực hiện function xử lí của **Consumer**.
* Với **pid > 0**, đây là ngữ cảnh của tiến trình cha. Thực hiện function xử lí của **Producer**.
* Trường hợp **pid < 0**, fork thất bại, thông báo lỗi và **return 1** (lỗi).

**A computer code with text

Description automatically generated with medium confidence**

*Hàm main: Đóng kết nối với shared memory và thu hồi bộ nhớ*

* Đóng kết nối với shared memory bằng **munmap**.
* Thu hồi bộ nhớ bằng **close**.

1. **Kết quả**

**A screen shot of a computer

Description automatically generated**

*Biên dịch file thực thi*

* Biên dịch file **BaiOT01.c** thành file thực thi **collatz**.
* Kiểm tra bằng lệnh **ls**, thấy file **collatz** có màu xanh (nghĩa là file này có quyền thực thi).

**A screen shot of a computer

Description automatically generated**

*Chạy file thực thi*

* Thực thi với input bằng **8** và input bằng **35**. Thấy kết quả giống như trong đề.

**A computer screen with text

Description automatically generated**

*Trường hợp lỗi*

* Các trường hợp không phải số nguyên dương được test ở trên bao gồm **-5** (số âm), **0** (số 0) và **hello** (một đoạn string bất kỳ).
* Khi nhập vào một số không phải số nguyên dương, chương trình sẽ xuất ra đoạn thông báo trên và dừng thực thi chương trình.