Phát triển chương trình bằng C cho hệ điều hành Linux cho phép bạn thực hiện tìm kiếm đệ quy các tệp, bắt đầu từ thư mục đã chỉ định, phù hợp với điều kiện trong Bảng. 3 và tùy chọn từ Bảng. 4.

Chương trình phải là một tiện ích bảng điều khiển, hoạt động của nó được định cấu hình bằng cách chuyển các đối số trong dòng khởi động và/hoặc sử dụng các biến môi trường (các tùy chọn là tùy chọn, bắt buộc phải có các đối số thư mục và search\_target):

lab11abcNXXXX [tùy chọn] thư mục search\_target

Chương trình phải thực hiện tìm kiếm đệ quy các tệp đáp ứng tiêu chí được chỉ định bởi đối số search\_target theo điều kiện trong Bảng. 3. Khi tìm thấy tệp phù hợp với tiêu chí tìm kiếm đã chỉ định, chương trình sẽ in đường dẫn đầy đủ đến tệp ở đầu ra tiêu chuẩn.

Khi các tùy chọn -h hoặc -v (hoặc các tùy chọn tương tự "dài" --help hoặc --version) được chỉ định, thông tin được chỉ định bởi tùy chọn sẽ được in và chương trình sẽ thoát. Các tùy chọn phải được chương trình hỗ trợ được hiển thị trong Bảng. 1.

Khi bạn xác định biến môi trường LAB11DEBUG, thông tin về nội dung được tìm thấy và vị trí trong tệp sẽ được in theo lỗi tiêu chuẩn (để dễ hiểu hơn tại sao tệp đáp ứng tiêu chí tìm kiếm) và mọi thông tin gỡ lỗi bổ sung có thể được in. Các biến môi trường phải được chương trình hỗ trợ được hiển thị trong Bảng. 2.

Tên chương trình phải bắt đầu bằng lab11, theo sau là hậu tố duy nhất cho biến thể. Hậu tố duy nhất được tạo thành từ các chữ cái đầu tiên của tên, họ (nếu có) và họ của sinh viên thực hiện công việc trong phòng thí nghiệm. Tiếp theo là số nhóm của học sinh. Các chữ cái Latinh viết thường và số Ả Rập (theo nghĩa truyền thống, tức là 0..9) được sử dụng. Ví dụ: nếu sinh viên làm thí nghiệm tên là Petr Sergeevich Ivanov, nhóm của anh ta là N32451, thì tên chương trình phải là lab11psiN32451.

Dự án (mã nguồn, tệp tiêu đề, Makefile và các tệp khác cần thiết cho việc lắp ráp) phải được chứa trong một thư mục riêng có cùng tên với tên chương trình (lab11abcNXXXXXX) và được xây dựng bằng tiện ích "make" tiêu chuẩn. Các tệp nguồn chương trình C phải được biên dịch bằng gcc. Makefile ít nhất phải hỗ trợ các mục tiêu all và clean. Nếu việc xây dựng dự án đòi hỏi nhiều hơn là thực hiện tất cả hoặc nếu cần có bất kỳ bước nào không tầm thường hoặc không rõ ràng để chạy và kiểm tra dự án thì phải thêm hướng dẫn xây dựng và chạy dự án vào tệp README.txt ở dạng văn bản thuần túy định dạng và đăng trong thư mục dự án.

Quy trình hoàn thành và nộp bài làm trong phòng thí nghiệm:

1. Chuẩn bị một bộ tệp kiểm tra (cả những tệp đáp ứng và những tệp không đáp ứng các tiêu chí tìm kiếm) và đặt chúng vào cây thư mục kiểm tra.

2. Hoàn thành nhiệm vụ, chuẩn bị tất cả các tệp dự án, biên dịch chương trình và thư viện với các cờ -Wall -Wextra -Werror và loại bỏ tất cả các cảnh báo và lỗi.

3. Kiểm tra chương trình trên nhiều thư mục khác nhau, đảm bảo không có lỗi, nếu không quay lại bước 2.

4. Chạy chương trình (tìm kiếm file trong cây thư mục test) chạy valgrind, đảm bảo không bị rò rỉ bộ nhớ. Nếu có rò rỉ, trước tiên hãy sửa chúng và quay lại bước 2. Nếu không có rò rỉ, hãy lưu báo cáo vào tệp valgrind.txt và thêm nó vào thư mục dự án.

5. Biên dịch chương trình với cờ -O3, kiểm tra lại chương trình bằng cách tìm kiếm trong nhiều thư mục khác nhau. Nếu tìm thấy lỗi hoặc cảnh báo, hãy quay lại bước 2.

6. Xóa tất cả các tệp thực thi và tệp trung gian khỏi thư mục dự án (làm sạch). Chỉ các tệp \*.c, \*.h, Makefile, README.txt, valgrind.txt mới được giữ lại trong kho lưu trữ.

7. Lưu trữ thư mục dự án bằng định dạng tar.gz. (tar -czvf lab11abcNXXXXX.tar.gz lab11abcNXXXXX/).

8. Gửi kho lưu trữ đã nhận đến email của giáo viên tiến hành phòng thí nghiệm (Girik A.V. tại itmo.osp@gmail.com, Gorlina A.V. tại gorlina.a.v@mail.ru, Grozov V.A. tại va\_groz@mail .ru), a lá thư với chủ đề “OSP LR11 Họ Tên Tên viết tắt NXXXXX”.

9. Đợi phản hồi qua thư hoặc trong bài học trong phòng thí nghiệm, loại bỏ mọi nhận xét có thể có (lặp lại từ điểm 1).

10. Nhận xác nhận từ giáo viên rằng công việc trong phòng thí nghiệm đã được hoàn thành thành công và sau đó chuẩn bị báo cáo dưới dạng điện tử (xem bố cục của báo cáo bên dưới).

11. Gửi kho lưu trữ phiên bản cuối cùng của dự án và báo cáo ở định dạng pdf (đừng quên chữ ký ở trang đầu tiên!) Đến email của giáo viên trong một bức thư với chủ đề “Báo cáo OSP về Họ LR11 Đầu tiên Tên từ viết tắt NXXXXX”. Tệp báo cáo phải được đặt tên là NXXXXX\_Last Name\_LR11.pdf.

12. Nhận một số câu hỏi về báo cáo từ giáo viên và đưa ra câu trả lời (hoặc có thể không nhận được nếu công việc thí nghiệm được thực hiện ở mức độ tốt và giáo viên không nghi ngờ gì về kiến ​​thức của học sinh). Nhận được xác nhận từ giáo viên rằng báo cáo đã được chấp nhận.

13. Thư giãn một chút và tiến hành công việc thí nghiệm tiếp theo :-)

Báo cáo phải được chuẩn bị ở định dạng pdf và bao gồm:

trang tiêu đề được thiết kế chính xác (có chữ ký của học sinh);

bài tập;

Makefile

báo cáo valgrind từ dòng «HEAP SUMMARY:»

ví dụ về hoạt động của chương trình (ảnh chụp màn hình);

mã nguồn của các chương trình có ý kiến.

Lưu ý 1. Khi thực hiện công việc trong phòng thí nghiệm, bạn nên sử dụng các chức năng của thư viện C tiêu chuẩn và các lệnh gọi hệ thống của hệ điều hành. Việc sử dụng C++ và do đó I/O kiểu C++ (các lớp ifstream/ofstream/...), vùng chứa và thuật toán STL (<string>, <vector>, <map>, ...) đều bị cấm.

Lưu ý 2. Chương trình phải có khả năng xử lý lỗi: trong trường hợp người dùng chỉ định tổ hợp tùy chọn không chính xác, chỉ định tệp không thể mở được, v.v. chương trình phải in thông báo chẩn đoán ra bảng điều khiển (lỗi tiêu chuẩn và/hoặc tệp nhật ký) trước khi thoát.

Lưu ý 3. Khi duyệt cây thư mục, bạn phải tính đến việc truy cập vào một số tệp và thư mục có thể không thành công và có lỗi (ví dụ: do thiếu quyền truy cập, v.v.). Trong trường hợp này, bạn nên in thông báo lỗi thành lỗi tiêu chuẩn và tiếp tục thu thập thông tin. Để đơn giản hóa việc thực hiện truyền tải, có thể bỏ qua các liên kết tượng trưng (và các phương tiện tương tự - liên kết cứng, liên kết gắn kết, v.v.).

Lưu ý 4. Nghiêm cấm sử dụng mảng tĩnh (có kích thước được chỉ định tại thời điểm biên dịch) cho bất kỳ dữ liệu nào có kích thước phụ thuộc vào dữ liệu đầu vào hoặc điều kiện khởi động. Để lưu trữ những dữ liệu đó, cần sử dụng bộ nhớ động và xác định dung lượng bộ nhớ cần thiết tùy theo tình huống. Mảng tĩnh có thể được sử dụng trong các tình huống đã biết kích thước tối đa của dữ liệu cần xử lý (và nó không vượt quá kích thước của ngăn xếp hoặc kích thước tối đa của mảng tĩnh được trình biên dịch cho phép).

Lưu ý 5. Thông báo thông tin được chương trình xuất ra luồng đầu ra tiêu chuẩn, thông báo lỗi - ra luồng lỗi tiêu chuẩn. Bằng cách xác định biến môi trường LAB11DEBUG, bạn có thể cho phép chương trình xuất thông báo gỡ lỗi sang luồng lỗi tiêu chuẩn.

Lưu ý 6: Nên sử dụng hàm getopt\_long() để xử lý các tùy chọn dòng lệnh. Nếu cần thiết, danh sách các tùy chọn có thể được mở rộng.

Lưu ý 7: Mặc dù bạn phải sử dụng trình biên dịch gcc để biên dịch chương trình, nhưng bạn nên tránh sử dụng các phần mở rộng GNU C nếu có thể và dựa vào việc sử dụng tiêu chuẩn C11 trở lên.

Lưu ý 8. Chương trình phải biên dịch và chạy thành công trên hệ điều hành Linux với phiên bản kernel >= 5.0. Nên tránh các chức năng và lệnh gọi hệ thống cụ thể của Linux và glibc bất cứ khi nào có thể và nên cố gắng tuân thủ POSIX. Nếu muốn và có thể, bạn nên kiểm tra hoạt động của chương trình không chỉ trên nền tảng x86\_64 GNU/Linux mà còn trên các nền tảng POSIX khác: x86/ARM/... + \*BSD/macOS/...

Bảng 1. Các tùy chọn dòng lệnh được chương trình hỗ trợ

| Tùy chọn | Mục đích |
| --- | --- |
| -v, --version | Hiển thị phiên bản chương trình và thông tin về chương trình (tên đầy đủ của người thực hiện, số nhóm, số phiên bản phòng thí nghiệm). |
| -h, --help | Hiển thị trợ giúp về các tùy chọn. |

Bảng 2. Các biến môi trường được chương trình hỗ trợ

| Biến đổi | Mục đích |
| --- | --- |
| LAB11DEBUG | Cho phép xuất thông tin gỡ lỗi. |

Bảng 3. Định dạng của đối số search\_target

| Tùy chọn (điều kiện) | Mục đích |
| --- | --- |
| Số chữ cái trong họ của học sinh thực hiện bài là số lẻ. | Một tìm kiếm được thực hiện cho chuỗi byte được chỉ định. Đối số search\_target có định dạng 0xhh[hh\*], trong đó hh là hai chữ số thập lục phân (số phải có tiền tố 0x).  Một ví dụ về chạy chương trình: ./lab11psiN32451 /home 0xc0ffee (Bắt đầu từ thư mục /home, tìm kiếm đệ quy được thực hiện đối với các tệp chứa chuỗi byte 0xc0 (192), 0xff (255) và 0xee (238).) |
| Số chữ cái trong họ của học sinh thực hiện bài làm là chẵn. | Một tìm kiếm được thực hiện cho chuỗi byte được chỉ định. Đối số search\_target có định dạng chuỗi được mã hóa UTF-8.  Ví dụ về chạy chương trình: ./lab11psiN32451 /home "lavender raf" (Bắt đầu từ thư mục /home, tìm kiếm đệ quy được thực hiện đối với các tệp chứa chuỗi byte 0xd0 0xbb ('l'), 0xd0 0xb0 ('a'), ..., 0xd1 0x84 ('f').) |

Bảng 4. Các hàm phải được sử dụng để thực hiện tìm kiếm tệp và thư mục

| Mã đề | Hàm |
| --- | --- |
| 1 | opendir()/readdir() không có đệ quy |