Zaawansowane programowanie w C++

Dr Michał Tanaś (http://mtanas.web.amu.edu.pl/)

Zakres tematyczny

1. Wstęp – kompilator GCC
2. Programowanie interfejsu graficznego – biblioteka QT
   a) Podstawy QT
   b) Widgety
   c) Mechanizm SIGNAL-SLOT

Zakres tematyczny

3. Programowanie wielowątkowe
   a) Podstawy
   b) Funkcje systemowe fork(), clone() i wait()
   c) Biblioteka pthreads
   d) System V IPC
   e) QThreads
4. Programowanie usług sieciowych
   a) Funkcje systemowe TCP/IP
   b) QSocket

GNU Compiler Collection (gcc)

GNU = GNU's Not Unix

FSF = Free Software Foundation

GPL = GNU Public License

LGPL = Lesser/Library GNU Public License

http://www.gnu.org/

http://gcc.gnu.org/

http://gcc.gnu.org/onlinedocs/

GPL = GNU Public License

- Zabronione jest zabranianie kopiowania
- Zabronione jest zabranianie modyfikacji, pod warunkiem wymienienia w dokumentacji orginalnego autora
- Obowiązkowe jest udostępnianie kodu źródłowego
- Program zawierający fragment kodu objętego licencją GPL/LGPL jest w całości objęty licencją GPL ("GPL infection").

LGPL = Lesser/Library GNU Public License

Program zawierający fragmenty kodu objętego licencją LGPL może nie być objęty licencją GPL jeżeli:

- Fragmenty te nie są integralną częścią kodu źródłowego programu ale są dołączane w czasie kompilacji (pliki nagłówkowe).
- Fragmenty kodu binarnego objętego licencją LGPL są dołączane przy pomocy standardowych mechanizmów wykorzystania bibliotek dynamicznych.

LGPL = Lesser/Library GNU Public License

Program zawierający fragmenty kodu objętego licencją LGPL może nie być objęty licencją GPL jeżeli:

- Fragmenty kodu binarnego objętego licencją LGPL nie są integralną częścią kodu binarnego programu (tzn. nie można linkować statycznie).
- Program nie jest rozpowszechniany łącznie z fragmentami kodu objętego licencją LGPL (tzn. program może wykorzystywać bibliotekę, ale nie może jej zawierać).
- Program będzie zgodny z wszystkimi przyszłymi wersjami wykorzystywanej biblioteki tak długo jak wersje te będą zgodne na poziomie interfejsu programisty.

gcc - sposob użycia

[gcc | g++] [opcje] -o wynik plik1 [plik2, plik3 , ...] [opcje]

gcc -Wall -O3 -o program1 plik1.cc plik2.cc -lpthread

g++ -Wall -O3 -o program1 plik1.cc plik2.cc -lpthread

gcc - pliki i katalogi

- pliki źródłowe: gcc plik1.cc plik2.cc plik3.cc ...
- nazwa pliku wynikowego: gcc -o plik

  domyślny plik wynikowy: a.out
- do łączenie biblioteki dynamicznej: gcc –lbiblioteka
- do łączenie biblioteki statycznej: gcc biblioteka.a
- dodatkowy katalog z plikami nagłówkowymi: gcc –Iścieżka
- dodatkowy katalog z bibliotekami: gcc -Lścieżka

gcc - zmienne środowiska

- dodatkowe katalogi z plikami nagłówkowymi:

  C_INCLUDE_PATH=ścieżka1:ścieżka2:...

  CPLUS INCLUDE PATH=ścieżka1:ścieżka2:...
- dodatkowe katalogi z bibliotekami dla kompilatora:

  LIBRARY_PATH=ścieżka1:ścieżka2:...
- dodatkowe katalogi z plikami dla programów:

  LD_LIBRARY_PATH=ścieżka1:ścieżka2:...

gcc - optymalizacja

- brak optymalizacji: gcc -O0
- optymalizacja która nie zwiększa czasu kompilacji: gcc -O1
- optymalizacja która nie zwiększa rozmiaru kodu: gcc -O2
- pełna optymalizacja: gcc -O3
- minimalizacja rozmiaru kodu: gcc -Os
- optymalizacja dla procesora:

  gcc -mtune=procesor

  gcc -mcpu=procesor
- wykorzystanie specyficznych instrukcji procesora:

  gcc -march=procesor

gcc - optymalizacja

- -march=generic - program będzie działać na wszystkich będących w użyciu procesorach rodziny x86. Obecnie równoważne -march=i686 (PentiumPro)
- -march=x86-64 - program będzie działać na każdym 64 bitowym procesorze rodziny x86
- -march=native - program będzie zoptymalizowany na procesor na którym jest kompilowany

gcc - optymalizacja

Używanie lub nie używanie rozszerzonych list rozkazów x86:

- -mmmx, -mno-mmx
- -msse, -mno-sse
- -msse2, -mno-sse2
- -msse3, -mno-sse3
- …
- -m3dnow, -mno-3dnow
- -mavx, -mno-avx
- -mavx2, -mno-avx2
- …

gcc - optymalizacja

Arytmetyka zmiennoprzecinkowa:

- -mfpmath=sse - przy pomocy SSE2. Opcja domyślna.
- -mfpmath=387 - przy pomocy instrukcji Intel 80387.
- -mfpmath=both - przy pomocy obu list instrukcji równocześnie - podwójny zestaw rejestrów i jednostek wykonawczych! Opcja odwiecznie eksperymentalna.

gcc - optymalizacja

Autowektoryzacja pętli:

- -ftree-vectorize - w łączenie autowektoryzacji. Wymaga co najmniej SSE.
- -ftree-vectorize-verbose=x - wyświetlanie w czasie kompilacji które pętle udało się zwektoryzować. Dla ciekawskich.

gcc - optymalizacja

Kod 32 lub 64 bitowy (dotyczy wyłącznie procesorów 64 bitowych). Wpływa na długość niektórych typów danych!

- -m32 – kod 32-bitowy
- -m64 – kod 64-bitowy.

gcc - opcje

- ostrzeżenia
  - gcc –Wostrzeżenie
  - zalecane: gcc –Wall
  - przestarzałe konstrukcje języka: gcc -Wno-deprecated
- define
  - gcc -Dnazwa
  - gcc -Dnazwa=wartość
- undef
  - gcc -Unazwa

gcc - opcje

- nazwa biblioteki

  gcc -Wl,-soname,libbiblioteka.so.wersja

- kod dla debugera

  gcc –g

- kod dla profilera

  gcc -p

gcc – etapy kompilacji

1. preprocesor: gcc –E

   wynik: plik źródłowy C/C++ bez dyrektyw preprocesora *.i, *.ii

2. kompilator: gcc –S

   wynik: plik źródłowy assemblera *.s

3. assembler: gcc –c

   wynik: plik "obiektowy"*.o

4. linker (tworzenie programu): gcc

   wynik: plik wykonywalny

   linker (tworzenie biblioteki): gcc –shared

   wynik: biblioteka dynamiczna

gcc – kompilacja bibliotek dynamicznych

- Przy kompilacji: kod w którym wszystkie adresy są niezależne od pozycji w pliku wykonywalnym

  gcc –fPIC

- Przy linkowaniu: polecenie tworzenia biblioteki:

  gcc -shared

- Nadanie nazwy bibliotece:

  gcc -Wl,-soname,libbiblioteka.so.x

Nazwy bibliotek

libnazwa.so.x.y.z

- obowiązkowy przedrostek: lib
- nazwa biblioteki: nazwa
- obowiązkowe rozszerzenie: .so.
- numer wersji: x.y.z

Nazwy bibliotek

libnazwa.so.x.y.z

Zmiana x oznacza:

- napisanie biblioteki "od zera"
- całkowita niezgodność z poprzednią wersją:
  - programy używające starej wersji nie mogą używać nowej
  - programy używające nowej wersji nie mogą używać starej
- możliwość zainstalowania wielu wersji różniących się x równocześnie (np. libqt5 i libqt6).

Nazwy bibliotek

libnazwa.so.x.y.z

Zmiana y oznacza:

- Rozszerzenie biblioteki o nową funkcjonalność (np. dodanie nowych funkcji).
- Zgodność "w dół" z poprzednią wersją:
  - programy używające starej wersji mogą używać nowej
  - programy używające nowej wersji niekoniecznie mogą używać starej

Nazwy bibliotek

libnazwa.so.x.y.z

Zmiana z oznacza:

- poprawienie błędów istniejących w poprzednich wersjach
- brak zmian w interfejsie programisty
- Zgodność "w dół" z poprzednią wersją:
  - programy używające starej wersji mogą używać nowej
  - programy używające nowej wersji mogą używać starej

Nazwy bibliotek

Pliki bibliotek:

- Plik na dysku:

  libnazwa.so.x.y.z

- Używany przez ld.so w czasie wykonywania programu: libnazwa.so.x
- Używany przez gcc w czasie kompilacji programu:

  libnazwa.so

Powiązania:

ln -s libbiblioteka.so.1.2.3 libbiblioteka.so.1.2

ln -s libbiblioteka.so.1.2 libbiblioteka.so.1

ln -s libbiblioteka.so.1 libbiblioteka.so

gcc - przykłady

Kompilacja programu:

gcc -Wall -O2 -c -o plik1.o plik1.c

gcc -Wall -O0 -c -o plik2.o plik2.c

g++ -Wall -O1 -c -o plik3.o plik3.cc

gcc -o program plik1.o plik2.o plik3.o -lmysqlclient -lstdc++ -lm

Kompilacja biblioteki:

gcc -Wall -O2 -fPIC -c -o plik1.o plik1.c

gcc -Wall -O0 -fPIC -c -o plik2.o plik2.c

g++ -Wall -O1 -fPIC -c -o plik3.o plik3.cc

gcc -shared -Wl,-soname,libbiblioteka.so.5 -o libbiblioteka.so.5.6.7 plik1.o plik2.o plik3.o -lmysqlclient -lstdc++ -lm

Biblioteki używane przez program - ldd

ldd ise

```
linux-vdso.so.1 (0x00007ffd8a379000)
libdl.so.2 => /lib/x86_64-linux-gnu/libdl.so.2 (0x00007f7ef42e2000)
libpthread.so.0 => /lib/x86_64-linux-gnu/libpthread.so.0
(0x00007f7ef42dd000)
libstdc++.so.6 => /lib/x86_64-linux-gnu/libstdc++.so.6 (0x00007f7ef4000000)
libm.so.6 => /lib/x86_64-linux-gnu/libm.so.6 (0x00007f7ef3f21000)
libgcc_s.so.1 => /lib/x86_64-linux-gnu/libgcc_s.so.1 (0x00007f7ef42bd000)
libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007f7ef3d40000)
/lib64/ld-linux-x86-64.so.2 (0x00007f7ef4303000)
```

Symbole w plikach binarnych - nm

nm spadaj

```
0000000000007310 r __GNU_EH_FRAME_HDR
                 U __gxx_personality_v0@CXXABI_1.3
0000000000003000 T _init
0000000000005000 R _IO_stdin_used
                 w _ITM_deregisterTMCloneTable
                 w _ITM_registerTMCloneTable
                 U __libc_start_main@GLIBC_2.34
0000000000003479 T main
                 U qt_version_tag@Qt_5.15
00000000000033f0 t register_tm_clones
0000000000003390 T _start
                 U strcmp@GLIBC_2.2.5
                 U strlen@GLIBC_2.2.5
```

Symbole w plikach binarnych - nm

Typy symboli:

A - zmienna absolutna

B - zmienna niezainicjalizowana

D - zmienna zainicjalizowana

R - stała

T - nazwa funkcji (zdefiniowanej w kodzie)

U - symbol niezdefiniowany (funkcja lub zmienna zewnętrzna)

W - słaby symbol niezdefiniowany

Symbole w plikach binarnych - nm

Słaby symbol niezdefiniowany

```
extern void podstawowa (int);
extern void zapasowa(int);
#pragma weak podstawowa=zapasowa
```

- Jeżeli funkcja podstawowa jest zdefiniowana (np. w dołączonej bibliotece) to jest wywoływana normalnie
- Jeżeli funkcja podstawowa nie jest zdefiniowana, to zamiast niej jest wywoływana funkcja zapasowa. Program daje się uruchomić mimo niezdefiniowanej funkcji.

Często stosowany gdy funkcja podstawowa wykorzystuje jakieś specyficzne możliwości sprzętu, a funkcja zapasowa pozwala na działanie programu na sprzęcie bez tych możliwości.

Symbole w plikach binarnych - strip

strip plik

- Usuwa z pliku wykonywalnego
  - tablice symboli
  - informacje dla debuggera
  - informacje dla profilera

Śledzenie wywołań funkcji systemowych - strace

strace spadaj

```
brk(0x55b9370ce000)                     = 0x55b9370ce000
brk(0x55b9370ef000)                     = 0x55b9370ef000
readlink("/proc/self/exe", "/home/mtanas/Documents/2023/buil"..., 4095) = 69
mmap(NULL, 733184, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f8c914e8000
sysinfo({uptime=45468, loads=[5152, 4448, 17920], totalram=8324354048, freeram=1291407360, sharedram=180498432, bufferram=156573696, totalswap=1023406080, freeswap=1023406080, procs=708, totalhigh=0, freehigh=0, mem_unit=1}) = 0
munmap(0x7f8c914e8000, 733184)          = 0
openat(AT_FDCWD, "/usr/share/fonts/truetype/noto/NotoSans-Regular.ttf", O_RDONLY) = 18
fcntl(18, F_SETFD, FD_CLOEXEC)
```