Лабораторная работа № U3. Журналы

В ОС (MS Windows и Unix) есть специальные системные журналы, в которых фиксируется информация о работе ОС и приложений, в том числе и прикладном ПО. Конечно, можно вести журнал в обычный текстовый файл с помощью самописных функций, но в ряде случаев удобнее пользоваться системной возможностью (в Unix это т.н. SysLog-сервер).

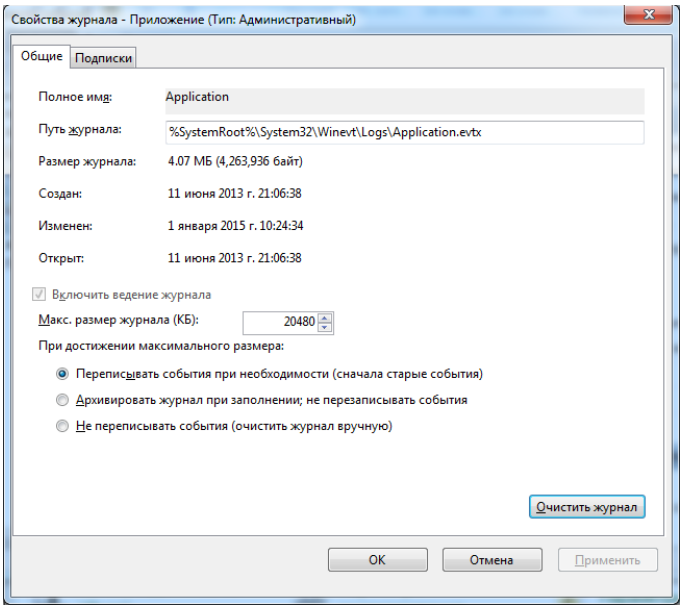
В Windows журналы встроены в ОС, доступ к журналам осуществляется через системные вызовы, например, просмотреть события можно через графический интерфейс или через PowerShell командой *Get-EventLog*, например ***Get-EventLog system***.

В Unix за журналы отвечает специальная программа-демон. Существует несколько версий – syslogd, syslog-ng, rsyslog и т.д. События могут фиксироваться как локальные так «поступившие» от других узлов локальной/глобальной сети, кроме того, syslog-демон имеет возможность фильтрации событий.

Примечание. Просмотр журнала можно осуществлять с помощью ***cat*** (отобразит все содержимое файла) или ***tail*** (отобразит последние N строк файла).

«Старые» записи из системных журналов могут быть автоматически удалены:

- в ОС Windows настраивается усечение журналов через графический интерфейс, примерно так:



- В ОС Unix Syslog функционалом усечения журналов не обладает, для этого используются дополнительное ПО – logrotate, запускаемый по расписанию.

  Настройка ротации – в /etc/logrotate.conf

  Запуск ротации вручную – ***logrotate /etc/logrotate.conf –v –f***

  (опции -v – вывод подробных сведений, -f принудительная ротация без контроля даты последнего запуска)

  Примечание. Для запуска нужны права системного администратора.

Запись в системном журнале(*syslog*) можно сделать с помощью программы ***logger*** (входит в один из системных пакетов, пример команды ***logger TESTMESSAGE***), либо с помощью системных вызовов:

- **openlog**(logPrefix, LOG_PID|LOG_CONS|LOG_NDELAY|LOG_NOWAIT, LOG_LOCAL0);

Первый параметр функции openlog() – префикс, который будет добавляться к каждой записи в системном журнале. Вторую опцию можно не указывать тогда там надо поставить 0, третий параметр тип журналирования, например openlog("mydaemon",LOG_PID,LOG_DAEMON)

- (void) **setlogmask**(LOG_UPTO(logLevel));

Функция setlogmask позволяет установить уровень приоритета сообщений, которые записываются в журнал событий. Если передать в качестве logLevel значение LOG_DEBUG, то в сочетании с макросом LOG_UPTO это означает, что в журнал будут записываться все сообщения с приоритетом, начиная с наивысшего и заканчивая LOG_DEBUG. (необязательная функция)

- **closelog**(); - закрывает журнал
- **syslog** (LogLevel, message);

где loglevel – приоритет события, а message - сообщение, подлежащее записи.

Например syslog(LOG_ALERT, "Database Error !");

в качестве message можно использовать конструкции аналогичные функции printf, например syslog(LOG_INFO, "Connection from host %d", callinghostname);

- Классы сообщений: LOG_USER, LOG_DAEMON, LOG_LOCAL0…LOG_LOCAL7
- Опции журнала: LOG_PID, LOG_CONS, LOG_PERROR
- Уровни приоритета: LOG_EMERG, LOG_ALERT, LOG_ERR, LOG_WARNING, LOG_INFO , LOG_DEBUG

Задание на самостоятельную работу:

1. Самостоятельно определить какого рода информация фиксируется в журналах из каталога /var/log/
2. Написать программу делающую запись в журнале. Например, слово hello

Доп.сведения:

- У исходника должно быть расширение .c
- Для использования журналов необходимо подключить библиотеку ***syslog.h***
- Для компилирования программы необходимо использовать команду gcc -o <программа> <исходник>, например если исходник называется mysource.c, то команда *gcc –o myproga mysource.c* создаст исполняемый файл myproga (расширение не важно)
- Для автоматического отслеживания и отображения на терминале добавляющихся строк в файле-журнале: команда tail с опцией -f