JURNALGA RO'YXATGA OLISH.

Yusupov Mirsaid Abdulaziz o'g'li
Farg'ona davlat universiteti amaliy matematikavainformatika kafedrasi katta o'qituvchisi
mirsaidbeky@gmail.com
Sahobiddinov Adhamjon Vohidjon o'g'li
Farg'ona davlat universiteti 2-bosqich talabasi
sahobiddinovadhamjon@gmail.com

Anotatsiya: *Ushbu maqolada axborot tizimlarida jurnalga ro'yxatga olish (logging) jarayoni, uning maqsadi, usullari va amaliy qo'llanilish sohalari keng yoritilgan. Jurnalga olish tizimlari tizimdagi hodisalarni qayd etish, nosozliklarni aniqlash, xavfsizlikni ta'minlash va tizim faoliyatini tahlil qilishda muhim rol o'ynaydi. Maqolada log yozuvlarining tuzilishi, ularni saqlash formatlari (masalan, JSON, XML, txt), log darajalari (info, warning, error, debug), log fayllarni aylantirish (log rotation) va ularni monitoring qilish vositalari tahlil qilingan.*

Kalit so'zlar: *Jurnalga olish, logging, log darajalari, xatolik, xavfsizlik, audit, Serilog, NLog, Log4j, Python logging, log rotation, tizim monitoringi, fayl formati, real vaqt, .NET*

Abstract : *This article explores the process of logging in information systems, its purpose, methods, and practical applications. Logging systems play a vital role in recording system events, detecting failures, ensuring security, and analyzing system performance. The article covers the structure of log entries, storage formats (e.g., JSON, XML, txt), log levels (info, warning, error, debug), log file rotation, and tools for log monitoring and analysis.*

Keywords: *Logging, log levels, error, security, audit, Serilog, NLog, Log4j, Python logging, log rotation, system monitoring, file format, real-time, .NET*

Аннотация: *В данной статье рассматривается процесс ведения журналов (логгирования) в информационных системах, его цели, методы и практическое применение. Системы логирования играют важную роль в фиксации событий, выявлении сбоев, обеспечении безопасности и анализе работы системы. В статье подробно рассматриваются структура лог-записей, форматы хранения (например, JSON, XML, txt), уровни логирования (info, warning, error, debug), ротация лог-файлов и инструменты мониторинга.*

Ключевые слова: *Логирование, уровни логов, ошибка, безопасность, аудит, Serilog, NLog, Log4j, Python logging, ротация логов, мониторинг системы, формат файла, реальное время, .NET*

Axborot texnologiyalarining jadal rivojlanishi va raqamli tizimlarning murakkablashib borishi bilan ularning ishonchli va xavfsiz ishlashini ta’minlash dolzarb masalaga aylandi. Aynan shu nuqtai nazardan, jurnalga ro‘yxatga olish (logging) texnologiyasi axborot tizimlarining ajralmas tarkibiy qismiga aylangan. Jurnalga olish – bu tizimda sodir bo‘ladigan voqealar, harakatlar va xatoliklar haqida ma’lumotlarni maxsus log fayllariga yoki ma’lumotlar bazasiga yozib borish jarayonidir. Bu jarayon dastur yoki tizimning faoliyatini nazorat qilish, yuzaga kelgan nosozliklarni tahlil qilish, xavfsizlikni ta’minlash va ishlash samaradorligini oshirish uchun xizmat qiladi. Tizimdagi har bir muhim harakat — foydalanuvchi kirishi, so‘rov yuborilishi, xizmat xatosi, ma’lumotlar o‘zgarishi — log yozuvi sifatida qayd etiladi. Ushbu yozuvlar dasturchilar va tizim administratorlari uchun diagnostika, audit va tahlil jarayonlarida asosiy manba hisoblanadi. Bugungi kunda jurnalga olish faqatgina muammolarni aniqlash vositasi emas, balki real vaqt monitoringi, xavfsizlik tahdidlarini aniqlash va

foydalanuvchi faoliyatini nazorat qilishda ham keng qo‘llaniladi. Yirik veb-ilovalar, mikroxizmatli arxitekturalar, moliyaviy tizimlar, tibbiyot platformalari kabi sohalarda loglar orqali nafaqat texnik, balki huquqiy va audit talablari ham bajariladi. Shu sababli jurnalga olish texnologiyasining to‘g‘ri tashkil etilishi, loglarni saqlash va tahlil qilish tizimlarining mavjudligi har qanday axborot tizimi samaradorligini oshirishda muhim rol o‘ynaydi. Ushbu maqolada jurnalga ro‘yxatga olishning asosiy tamoyillari, ishlash mexanizmlari, mavjud vositalar va ularning amaliy qo‘llanilishi batafsil yoritiladi.

Jurnalga olish (logging) — bu axborot tizimida yuz beradigan muhim voqealarni yozib borish jarayonidir. Bu jarayon orqali tizimdagi holatlar, foydalanuvchi harakatlari, xatoliklar yoki ogohlantirishlar maxsus fayllar yoki ma’lumotlar bazasida saqlanadi. Jurnalga olishning asosiy maqsadi — tizimdagi har bir harakatni izchil qayd etib borish orqali tizimni nazorat qilish, nosozliklarni aniqlash, xavfsizlikni ta’minlash va tizim samaradorligini oshirishdan iborat. Dasturiy ta'minot ishlab chiqishda loglar odatda diagnostika vositasi sifatida qo‘llaniladi, ya'ni ishlab chiquvchi yoki administrator log fayllar orqali tizimning qayerida muammo yuzaga kelganini aniqlay oladi.

Jurnal yozuvlari turli darajalarda yaratiladi. Asosiy log darajalari quyidagilardir: *Debug* – dasturchilar uchun ishlab chiqish vaqtida zarur bo‘lgan batafsil ma’lumotlarni beradi; *Info* – tizimdagi umumiy holat yoki muvaffaqiyatli bajarilgan amallar haqida axborot beradi; *Warning* – potentsial muammolar haqida ogohlantiradi; *Error* – bajarilmagan amallar yoki xatoliklar haqida ma’lumot beradi; *Critical* – tizimda jiddiy nosozlik yuzaga kelganligini bildiradi. Ushbu darajalar log ma’lumotlarini to‘g‘ri tahlil qilishda muhim rol o‘ynaydi.

Jurnalga olishning ishlash prinsipi shundan iboratki, tizimda har qanday muhim hodisa sodir bo‘lishi bilan u haqidagi ma’lumot dasturiy log moduli orqali maxsus log yozuvi sifatida yaratilib, belgilangan formatda faylga yoziladi yoki log serverga yuboriladi. Bu yozuvlar odatda vaqt tamg‘asi (timestamp), hodisa darajasi

(level), hodisa manbai (source) va hodisa matnidan (message) iborat bo‘ladi. Log yozuvlari turli formatlarda saqlanishi mumkin, eng ko‘p qo‘llaniladiganlari — matn fayli (txt), JSON, XML formatlari. Yuqori yuklama ostida ishlovchi yoki tarqatilgan tizimlar uchun loglarni to‘g‘ri tashkil etish muhim ahamiyatga ega. Shu sababli log fayllarni avtomatik aylantirish (log rotation), siqish, arxivlash va eskilarini o‘chirish mexanizmlari ham joriy qilinadi. Bundan tashqari, zamonaviy tizimlarda loglarni real vaqt rejimida kuzatish, markazlashgan serverda to‘plash va tahlil qilish uchun maxsus monitoring tizimlari, masalan, ELK (Elasticsearch, Logstash, Kibana) steki yoki Splunk kabi vositalar qo‘llaniladi. Shunday qilib, jurnalga olish nafaqat texnik, balki xavfsizlik, auditori va tahliliy jihatdan ham axborot tizimining ajralmas elementi hisoblanadi.

Jurnalga olish turlari va texnik yondashuvlar tizim arxitekturasiga, foydalanuvchilarning ehtiyojlariga va konkret ilovaning xususiyatlariga qarab farq qiladi. Jurnalga olishning eng keng tarqalgan turlari — lokal (local) va markazlashtirilgan (centralized) loglash tizimlaridir. Mahalliy loglash tizimida har bir server yoki ilova o‘zining log fayllarini mustaqil tarzda yaratadi va saqlaydi. Bu usul kichik yoki bir necha komponentlardan iborat tizimlarda samarali bo‘lishi mumkin, chunki barcha loglar bitta mashinada saqlanadi va tizim ma'lumotlarini tahlil qilish nisbatan osonroqdir. Biroq, katta tizimlar va tarqatilgan ilovalarda bunday usul o‘ta murakkab va samarali bo‘lmasligi mumkin. Shuning uchun markazlashtirilgan loglash tizimlari ko‘proq ishlatiladi, bunda barcha log yozuvlari markaziy serverga yuboriladi. Bu usul loglarni tahlil qilishni osonlashtiradi, barcha tizim holatini bir joydan kuzatib borish imkonini beradi va ko‘plab serverlar orasidagi bir xil siyosatni amalga oshirishni ta’minlaydi.

Jurnalga olishni tashkil etishning yana bir muhim yondashuvi — real vaqt monitoringidir. Ushbu texnik yondashuvda loglar darhol tizimda sodir bo‘lgan voqealar haqida ma’lumot beradi. Real vaqt monitoringi tizim administratorlariga darhol xatoliklarni aniqlash, xavfsizlik tahdidlarini oldini olish va tizimning

holatini uzluksiz nazorat qilish imkonini yaratadi. Bunda loglar maxsus tahlil vositalari orqali real vaqtda qayd etiladi, masalan, ELK stekida loglarni to‘plash va tahlil qilish uchun Elasticsearch, Logstash va Kibana kombinatsiyasi ishlatiladi. Bu yondashuv yuqori darajadagi tarmoq xavfsizligini ta'minlash va xizmat ko‘rsatishning uzluksizligini nazorat qilish uchun zarurdir.

Log fayllarni aylantirish (log rotation) ham muhim texnik yondashuvdir, ayniqsa, tizimda doimiy ravishda katta miqdordagi ma’lumotlar yig‘ilayotgan bo‘lsa. Log fayllari ma’lum hajmga yetganda yoki belgilangan vaqt o'tgach avtomatik tarzda yangi faylga o‘tkaziladi, bu eski fayllarni arxivlash va yangi log fayllarini yaratish jarayonini osonlashtiradi. Log aylantirish tizimi log fayllarining o‘sishini cheklashga yordam beradi va tizimning doimiy ishlashini ta'minlaydi. Bunda eski loglar saqlanib, kerak bo‘lganda tahlil qilish uchun saqlanadi, yangi loglar esa yangi faylga yoziladi. Bundan tashqari, loglarni tahlil qilish va qayta ishlash uchun maxsus vositalar, masalan, Splunk, Fluentd yoki Graylog kabi tizimlar ham mavjud. Bu tizimlar log ma'lumotlarini yig‘ish, tahlil qilish va vizualizatsiya qilishda yordam beradi. Markazlashtirilgan tizimlar ko‘pincha loglarni to‘plash va tahlil qilish uchun bulutli xizmatlardan ham foydalanadi. Bulutli loglash tizimlari, masalan, Amazon CloudWatch, Google Stackdriver yoki Azure Monitor kabi xizmatlar, loglarni saqlash va tahlil qilishni samarali qilish imkoniyatini ta’minlaydi. Bu tizimlar o‘zaro integratsiyalashgan holda ishlaydi va katta tizimlarda loglarni boshqarishni ancha soddalashtiradi. Shunday qilib, jurnalga olish turlari va texnik yondashuvlar tizimning miqyosi, ehtiyojlari va xavfsizlik talablari asosida tanlanadi. Har bir yondashuvning o‘zining afzalliklari va cheklovlari mavjud, shuning uchun tizim arxitekturasi va qo‘llanilish sohasiga qarab to‘g‘ri tanlov qilish zarur.

Jurnalga olish tizimlarida log kutubxonalari va vositalari juda muhim ahamiyatga ega, chunki ular loglarni yaratish, saqlash, tahlil qilish va kuzatish jarayonlarini soddalashtiradi. Hozirgi kunda bir qancha mashhur log kutubxonalari

mavjud bo‘lib, ular o‘zlarining imkoniyatlari, konfiguratsiyasi va qo‘llanishi bilan ajralib turadi. Ulardan eng keng tarqalganlari Serilog, NLog, Log4j, Python logging moduli va Logbackdir.

Serilog — bu .NET platformasida ishlaydigan zamonaviy va qulay log kutubxonasidir. Serilogning asosiy afzalligi uning strukturaviy loglarni yaratish imkoniyatidir. Bu kutubxona loglarni JSON formatida yoki boshqa strukturaviy formatlarda saqlashni qo‘llab-quvvatlaydi, bu esa log ma’lumotlarini keyinchalik tahlil qilishni osonlashtiradi. Serilog, shuningdek, loglarni turli manbalar orqali yozib olishni qo‘llab-quvvatlaydi, masalan, fayl tizimi, ma’lumotlar bazalari, bulutli xizmati yoki real vaqt monitoringi tizimlari. Bu kutubxona tarmoq xizmatlarida keng qo‘llaniladi, chunki uning konfiguratsiyasi va integratsiyasi juda qulay.

NLog — bu .NET platformasida ishlaydigan yana bir mashhur log kutubxonasidir. NLog o‘zining konfiguratsiya imkoniyatlari va yuqori darajadagi moslashuvchanligi bilan tanilgan. U log fayllarini turli formatlarda saqlashga imkon beradi, masalan, XML, JSON yoki matnli fayllar. NLog xatoliklarni qayd etishda, tizim monitoringida va auditing jarayonlarida samarali qo‘llaniladi. Bu kutubxona tizimni tahlil qilish uchun samarali vosita bo‘lib, foydalanuvchining ish faoliyatini to‘liq kuzatib borish imkonini beradi.

Log4j — Java dasturlash tili uchun eng mashhur log kutubxonalaridan biridir. Log4j o‘zining kuchli va kengaytirilgan konfiguratsiyasi bilan ajralib turadi, shuningdek, u loglarni real vaqt rejimida yaratish va tahlil qilish uchun zarur vositalarni ta’minlaydi. Log4j log yozuvlarini turli xil formatlarda saqlashni qo‘llab-quvvatlaydi va serverlarning bir nechta loglarini markazlashtirilgan tizimda to‘plash imkonini beradi. Hozirda Log4j turli yirik tizimlarda, jumladan, veb-ilovalarda va mikroxizmatli arxitekturalarda keng qo‘llaniladi.

Python logging moduli esa Python dasturlash tilida log yozuvlarini yaratish uchun keng qo‘llaniladi. Bu kutubxona juda oddiy va moslashuvchan bo‘lib, uni

ishlab chiquvchilar oddiy dasturni debugging qilishdan tortib, murakkab tizimlarni monitoring qilishgacha turli maqsadlarda ishlatishlari mumkin. Python logging moduli log darajalarini sozlash, faylga yozish va loglarni tahlil qilishda foydalidir. Shuningdek, bu kutubxona boshqa kutubxonalar va tizimlar bilan integratsiya qilish imkonini beradi, masalan, Splunk yoki ELK steki bilan.

Logback — Java platformasida ishlovchi yana bir kuchli log kutubxonasi bo‘lib, Log4j ning keyingi avlodi sifatida yaratilgan. Logback yuqori samaradorlik va yaxshi konfiguratsiya imkoniyatlarini taqdim etadi. U loglarni turli darajalarda yozib olishni qo‘llab-quvvatlaydi va tizimning yukini minimallashtiradi. Logback yordamida real vaqt rejimida monitoring va log fayllarni saqlash osonlashadi. Ushbu kutubxona yirik va murakkab tizimlar, shuningdek, tizimni audit qilish va xavfsizlikni ta’minlash jarayonlarida keng qo‘llaniladi.

Yuqoridagi kutubxonalar loglarni samarali tarzda yaratish, saqlash va tahlil qilish imkonini beradi, bu esa tizimning ishlashini yaxshilash, xavfsizlikni ta’minlash va nosozliklarni aniqlashda juda muhim ahamiyatga ega. Har bir kutubxona o‘zining xususiyatlariga va tizim ehtiyojlariga qarab tanlanadi, masalan, Serilog va NLog .NET platformalarida, Log4j va Logback esa Java platformalarida keng qo‘llaniladi. Python logging esa Python dasturchilari uchun samarali va qulay vosita hisoblanadi. Shu tarzda, log kutubxonalari yordamida tizimlar o‘zlarining holatlarini to‘liq kuzatib borish va muammolarni tezda aniqlash imkoniyatiga ega bo‘ladi.

Jurnalga ro‘yxatga olish (logging) texnologiyalari axborot tizimlarining ishonchli ishlashini ta'minlash, tizim monitoringi, xavfsizlikni boshqarish va nosozliklarni aniqlashda muhim rol o‘ynaydi. Ushbu maqolada jurnalga olish jarayonining asosiy tushunchalari, ishlash prinsiplari, turlari va mashhur log kutubxonalari haqida batafsil ma’lumot berildi. Tizimlarni samarali boshqarish uchun loglar faqatgina xatoliklar yoki nosozliklar haqida ma’lumot bermaydi, balki tizimning umumiy holatini, foydalanuvchi harakatlarini va xavfsizlik holatini tahlil

qilish imkonini ham beradi. Loglar yordamida tizim faoliyatining barcha jihatlari nazorat qilinadi va real vaqt monitoringi orqali yuzaga kelgan muammolar tezda aniqlanadi. Biroq, loglarni to‘g‘ri va samarali tashkil etish uchun ularning tuzilishini, formatlarini va saqlash metodlarini yaxshi o‘rganish zarur. Har bir tizim uchun loglarni yaratish va saqlash jarayoni individual yondashuvni talab qiladi, bu esa tizimning o‘ziga xos xususiyatlariga, hajmiga va ishlash shartlariga bog‘liq. Shuning uchun loglarni to‘g‘ri tashkil etish, markazlashtirilgan monitoring tizimlari va bulutli xizmatlardan foydalangan holda loglarni tahlil qilish tizim samaradorligini oshiradi va tizim administratorlarining ishini sezilarli darajada osonlashtiradi. Kelib chiqadigan takliflar, birinchidan, log fayllarini markazlashtirilgan tarzda yig‘ish va ularni samarali tahlil qilishni ta'minlash uchun zamonaviy vositalarni joriy etishdir. Misol uchun, ELK stekini yoki Splunk kabi vositalar yordamida loglarni to‘plash va tahlil qilish tizimning tezkor ishlashini ta'minlaydi. Ikkinchidan, log darajalarini va log aylantirish (log rotation) mexanizmlarini to‘g‘ri sozlash tizimni oson boshqarish va muammolarni aniqlash imkoniyatlarini beradi. Uchinchidan, loglarni real vaqt rejimida monitoring qilish uchun maxsus xizmatlardan foydalanish, tizimdagi tahdidlarni tezda aniqlash va xavfsizlikni ta'minlashda katta yordam beradi. Nihoyat, log fayllarining saqlash va saqlash muddatini o‘rganish, ular haqida ma’lumotlarni muntazam ravishda arxivlash va tahlil qilish tizim xavfsizligini kuchaytirishga yordam beradi.

Jurnalga olish tizimlarini rivojlantirishda yirik tizimlar va tarqatilgan arxitekturalarda loglarni markazlashgan tarzda boshqarish va tahlil qilish zarurligini inobatga olish lozim. Loglar orqali tizim holatining har bir aspektini kuzatib borish, xatoliklarni aniqlash va xavfsizlik tahdidlariga qarshi choralar ko‘rish imkoniyatlarini oshiradi. Shu sababli, jurnalga olishning barcha imkoniyatlarini to‘liq foydalangan holda tizimlar samarali boshqarilishi kerak.

Foydalanilgan adabiyotlar

1. Microsoft Docs, "Logging in ASP.NET Core", docs.microsoft.com, 2024.

2. Apache Logging Services, "Log4j 2 Documentation", logging.apache.org, 2024.

3. Serilog Documentation, serilog.net, 2024.

4. NLog Documentation, nlog-project.org, 2024.

5. Python Logging HOWTO, Python Docs, docs.python.org, 2024.

6. Marko Sillanpää, *Logging in Distributed Systems*, Springer, 2019.

7. Chris Richardson, *Microservices Patterns*, Manning Publications, 2019.

8. Mamirovich, I. S., Revkatovich, I. E., Rustamjon o‘g, H. O. K., & Yigitali o'g'li, R. J. (2023). IJTIMOIY TARMOQLARDA BIG DATA TEXNOLOGIYASIDAN FOYDALANISH TAHLILI. *" RUSSIAN" ИННОВАЦИОННЫЕ ПОДХОДЫ В СОВРЕМЕННОЙ НАУКЕ*, *9*(1).

9. Nurmamatovich, T. I. (2024, April). SUN’IY NEYRONNING MATEMATIK MODELI HAMDA FAOLLASHTIRISH FUNKTSIYALARI. In " USA" INTERNATIONAL SCIENTIFIC AND PRACTICAL CONFERENCE TOPICAL ISSUES OF SCIENCE (Vol. 17, No. 1).

10. Memcached Developers. (2023). *Memcached Official Documentation*. [https://memcached.org]

11. Zhang, W., & Liu, X. (2020). *Performance Analysis of Cache Systems in Cloud Computing*. IEEE Access.

12. Apache Software Foundation. (2023). *Caching in Apache HTTP Server*. [https://httpd.apache.org]