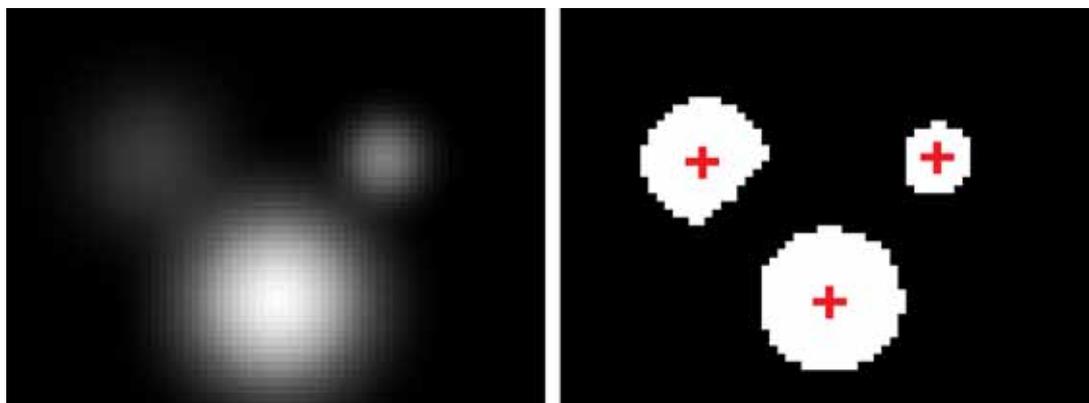


Результаты сегментации приведены на рисунке. Чувствительность распознавания объектов для данного изображения принята равной 50 %.

деры услуг, а не пользователь. В-четвертых, доступ к данным предоставляется из любой точки планеты, где есть выход в сеть, и с любого устройства, будь то



Исходное изображение (слева) и результаты сегментации (справа)

В результате проведенного исследования разработан и реализован в виде программы для ПК алгоритм сегментации полноградационных изображений.

**Список литературы**

1. Хорн Б.К.П. Зрение роботов. – М.: Мир, 1989. – 487 с.
2. Сальников И.И. Растровые пространственно-временные сигналы в системах анализа изображений. – М.: Физматлит, 2009. – 248 с.
3. Мартышкин А.И. Программный комплекс для имитационного моделирования диспетчеров задач многопроцессорных систем с использованием приоритетных сетей массового обслуживания // Фундаментальные исследования. – 2014. – № 11-10. – С. 2155-2159.

**ИСПОЛЬЗОВАНИЕ ОБЛАЧНЫХ ТЕХНОЛОГИЙ  
ДЛЯ БИЗНЕСА**

Калягин И.Н., Воронцов А.А.

Пензенский государственный технологический университет, Пенза, e-mail: alexey314@ya.ru

За последние годы одной из наиболее популярных тем в области информационных технологий стали «облака». Внедрение облачных технологий стремительно происходит в различных сферах деятельности: медицине, образовании, органах государственной власти, науке, сфере развлечений и бизнесе.

Цель данной статьи – рассмотреть использование облачных технологий в малом и среднем бизнесе, проанализировать их преимущества и недостатки.

Облачные технологии представляют собой способ распределённой обработки данных, в котором компьютерные ресурсы и мощности предоставляются пользователю как Интернет-сервис [1, 2]. Иначе говоря, это электронное хранилище данных пользователя в сети Интернет, которое позволяет хранить, редактировать, пользоваться нужными программами и сервисами с любого доступного ему компьютера или смартфона, а также делиться интересными файлами и документами с друзьями и коллегами.

Сфера бизнеса быстро оценила преимущества, предоставляемые облачными технологиями. В-первых, их использование позволяет снизить расходы на покупку и модернизацию программного обеспечения и оборудования. Во-вторых, пользователь оплачивает фактическое использование услуги, то есть только за то время и тот набор функций, который ему необходим. В-третьих, снижаются затраты на техническое обслуживание и обновление программного обеспечения, поскольку этим занимаются провай-

дер компьютер, планшетный компьютер или смартфон. Помимо этого, облачные технологии позволяют легко делиться информацией с другими людьми, например, с клиентами и сотрудниками.

Однако существуют и недостатки при использовании облачного сервиса [3]. В первую очередь, это вопросы безопасности, конфиденциальности и сохранения данных, по поводу чего существует огромное количество споров. Во-вторых, предприятие оказывается в «технологической зависимости» от провайдера, и если встанет вопрос о его смене, то, скорее всего, возникнут сложности и различные нюансы. И не стоит забывать об актуальном для российских пользователей недостатке, пожалуй, самом главном, когда речь идет об облачных технологиях, – это надежный и быстрый доступ в сеть Интернет.

Какие же ресурсы предлагают владельцам малого бизнеса провайдеры облачных технологий? Самые разные, начиная от базового набора в виде почты, календаря, sms-напоминаний о встречах, до решений для автоматизации магазина.

На сегодняшний день облачные технологии используются и на мобильных платформах: смартфонах и планшетных компьютерах. Все операторы Большой тройки (МТС, Билайн, Мегафон) предлагают своим абонентам облачные пакеты, рассчитанные на любые потребности корпоративных пользователей. Наиболее полный набор облачных услуг для абонентов предлагает компания МТС. Ее решения включают в себя, как стандартные пакеты от Microsoft и Google, так и собственные разработки для отдельных направлений бизнеса: транспортных и строительных компаний, предприятий оптовой и розничной торговли, охранных предприятий, а также служб инкассации. Имея оборудование с SIM-картой МТС, абоненты могут подключить по выгодным тарифам и за минимальную абонентскую плату необходимые облачные приложения, в их числе: Office 365, пакеты 1С (1С-предприятие, 1С-управление небольшой фирмой WEB, 1С управление торговлей), REGBERRY – специальный облачный сервис от МТС, который послужит отличным помощником в оформлении любых необходимых документов при создании фирмы; Небо – облачная бухгалтерия, с помощью которой возможно проведение любых операций, в том числе ведение бухгалтерского учета любой сложности, загрузка выписок из клиент-банка,

расчет заработной платы, формирование отчетностей и деклараций; Спонтания – уникальный онлайн-ресурс для проведения видеоконференций, главной особенностью которого является возможность использования с различных устройств, работающих на платформе Android, IOS [4]. С его помощью возможно проведение переговоров из любых стран мира в реальном времени.

Облачные технологии предоставляют предприятиям малого и среднего бизнеса возможность повышения эффективности своей работы, при этом снизив финансовые затраты на улучшение программного обеспечения, подобрав тарифный план с необходимым набором функций.

#### Список литературы

1. Облачные вычисления [Электронный ресурс]. – URL: [https://ru.wikipedia.org/wiki/Облачные\\_вычисления](https://ru.wikipedia.org/wiki/Облачные_вычисления) (Дата обращения: 20.12.2015).
2. Во что обходится миграция систем в облака? [Электронный ресурс]// CNews: Издание о высоких технологиях. – URL: <http://www.cnews.ru>. (Дата обращения: 20.12.2015).
3. Мартышкин А.И. Исследование алгоритмов планирования процессов в системах реального времени // Современные методы и средства обработки пространственно-временных сигналов: сборник статей XIII Всероссийской научно-технической конференции / Под ред. И.И. Сальникова. – Пенза, 2015. – С.118-124.
4. Бершадская Е.Г., Зубков А.В. Оценка возможностей моделирования производственным систем // Международный студенческий научный вестник. 2015, – № 3-2, – С. 265-266.

### ОБЗОР СРЕДСТВ УПРАВЛЕНИЯ ПРОЦЕССАМИ И РЕСУРСАМИ МНОГОПРОЦЕССОРНЫХ ОПЕРАЦИОННЫХ СИСТЕМ

Карасева Е.А., Мартышкин А.И.

Пензенский государственный технологический университет, Пенза, e-mail: [alexey314@ya.ru](mailto:alexey314@ya.ru)

В качестве примера рассмотрим семафоры, мьютексы и мониторы.

В 1965 году Дейкстра предложил использовать переменную для подсчета сигналов запуска, сохраненных на будущее [1]. Им был предложен новый тип переменных – **семафоры**, значение которых может быть нулем (в случае отсутствия сохраненных сигналов активизации) или некоторым положительным числом, соответствующим количеству отложенных активизирующих сигналов.

Дейкстра предложил две операции, *down* и *up* (обобщения *sleep* и *wakeup*). Операция *down* сравнивает значение семафора с нулем. Если значение семафора больше нуля, операция *down* уменьшает его и просто возвращает управление. Если значение семафора равно нулю, процедура *down* не возвращает управление процессу, а процесс переводится в состояние ожидания. Все операции проверки значения семафора, его изменения и перевода процесса в состояние ожидания выполняются как единое и неделимое **элементарное действие**. Тем самым гарантируется, что после начала операции ни один процесс не получит доступа к семафору до окончания или блокирования операции. Элементарность операции чрезвычайно важна для разрешения проблемы синхронизации и предотвращения состояния состязания.

Операция *up* увеличивает значение семафора. Если с этим семафором связаны один или несколько ожидающих процессов, которые не могут завершить более раннюю операцию *down*, один из них выбирается системой (например, случайным образом) и ему разрешается завершить свою операцию *down*. Таким образом, после операции *up*, примененной к семафору, связанному с несколькими ожидающими процессами, значение семафора так и останется равным 0, но число ожидающих процессов уменьшится на единицу. Операция увеличения значения семафора и активизации процесса тоже неделима. Ни один процесс не

может быть заблокирован во время выполнения операции *up*, как ни один процесс не мог быть заблокирован во время выполнения операции *wakeup* в предыдущей модели [1].

Иногда используется упрощенная версия семафора, называемая мьютексом [1]. Мьютекс не способен считать, он может лишь управлять взаимным исключением доступа к совместно используемым ресурсам или кодам. Реализация мьютекса проста и эффективна, что делает использование мьютексов особенно полезным в случае потоков, действующих только в пространстве пользователя. **Мьютекс** – переменная, которая может находиться в одном из двух состояний: заблокированном или не заблокированном. Поэтому для описания мьютекса требуется всего один бит, хотя чаще используется целая переменная, у которой 0 означает не заблокированное состояние, а все остальные значения соответствуют заблокированному состоянию. Значение мьютекса устанавливается двумя процедурами. Если поток (или процесс) собирается войти в критическую область, он вызывает процедуру *mutexlock*. Если мьютекс не заблокирован (то есть вход в критическую область разрешен), запрос выполняется и вызывающий поток может попасть в критическую область [1, 2].

Напротив, если мьютекс заблокирован, вызывающий поток блокируется до тех пор, пока другой поток, находящийся в критической области, не выйдет из нее, вызвав процедуру *mutex\_unlock*. Если мьютекс блокирует несколько потоков, то из них случайным образом выбирается один.

В случае потоков ситуация кардинально меняется, поскольку нет прерываний по таймеру, останавливающих слишком долго работающие потоки. Поток, пытающийся получить доступ к семафору и находящийся в состоянии активного ожидания, заикнется навсегда, поскольку он не позволит предоставить процессор другому потоку, желающему снять блокировку.

Если два или больше процессов разделяют частично или полностью адресное пространство, различие между процессами и потоками частично размывается, но тем не менее все равно остается. Два процесса с общим адресным пространством все равно обладают разными открытыми файлами, аварийными таймерами и прочими характеристиками, присущими процессам, в то время как два потока, разделяющие адресное пространство, разделяют и все остальное. И в любом случае несколько процессов, совместно использующих адресное пространство, никогда не будут столь же эффективны, как потоки на уровне пользователя, поскольку управление потоками всегда происходит через ядро.

Чтобы упростить написание программ, в 1974 году Хоар и Бринч Хансен предложили примитив синхронизации более высокого уровня, называемый **монитором** [1]. Монитор – набор процедур, переменных и других структур данных, объединенных в особый модуль или пакет. Процессы могут вызывать процедуры монитора, но у процедур, объявленных вне монитора, нет прямого доступа к внутренним структурам данных монитора.

Реализации взаимных исключений способствует важное свойство монитора: при обращении к монитору в любой момент времени активным может быть только один процесс. Мониторы являются структурным компонентом языка программирования, поэтому компилятор знает, что обрабатывать вызовы процедур монитора следует иначе, чем вызовы остальных процедур. Обычно при вызове процедуры монитора первые несколько команд процедуры проверяют, нет ли в мониторе активного процесса. Если активный