

```

        extract_number_and_incr(destination, source) int
        *destination; unsigned char **source; { extract_number_and_incr(destination, *source); *source += 2; } #ifndef EXTRACT_MACROS #undef EXTRACT_NUMBER_AND_INCR #define EXTRACT_NUMBER_AND_INCR(dest, src) extract_number_and_incr (&dest, &src) #endif /* not EXTRACT_MACROS */ #endif /* DEBUG */ /* If DEBUG is defined, Regex prints many voluminous messages about what it is doing (if the variable 'debug' is nonzero). If linked with the main program in 'iregex.c', you can enter patterns and strings interactively. And if linked with the main program in 'main.c' and the other test files, you can run the already-written tests. */ #ifdef DEBUG /* We use standard I/O for debugging. */ #include <stdio.h> /* It is useful to test things that "must" be true when debugging. */ #include <assert.h> static int debug = 0; #define DEBUG_STATEMENT(e) #define DEBUG_PRINT1(x) if (debug) printf (x) #define DEBUG_PRINT2(x1, x2) if (debug) printf (x1, x2) #define DEBUG_PRINT3(x1, x2, x3) if (debug) printf (x1, x2, x3) #define DEBUG_PRINT4(x1, x2, x3, x4) if (debug) printf (x1, x2, x3, x4) #define DEBUG_PRINT_COMPILED_PATTERN(p, s, e) if (debug) print_partial_compiled_pattern (s, e) #define DEBUG_PRINT_DOUBLE_STRING(w, s1, sz1, s2, sz2) \ if (debug) print_double_string (w, s1, sz1, s2, sz2) extern void printchar(); /* Print the fastmap in human-readable form. */ void print_fastmap (fastmap) char *fastmap; { unsigned was_a_range = 0; unsigned i = 0; while (i < (1 << BYTEWIDTH)) { if (fastmap[i++] { was_a_range = 0; printchar (i - 1); while (i < (1 << BYTEWIDTH) && fastmap[i]) { was_a_range = 1; i++; } if (was_a_range) { printf ("-"); printchar (i - 1); } } putchar ('\n'); } /* Print a compiled pattern string in human-readable form, starting at the START pointer into it and ending just before the pointer END. */ void print_partial_compiled_pattern (start, end) unsigned char *start; unsigned char *end; { int mcnt, mcnt2; unsigned char *p = start; unsigned char *pend = end; if (start == NULL) { printf ("(null)\n"); return; } /* Loop over pattern commands. */ while (p < pend) { switch ((re_opcode_t) *p++) { case no_op: printf ("/no_op"); break; case exactn: mcnt = *p++; printf ("/exactn/%d", mcnt); do { putchar ('/'); printchar (*p++); } while (--mcnt); break; case start_memory: mcnt = *p++; printf ("/start_memory/%d/%d", mcnt, *p++); break; case stop_memory: mcnt = *p++; printf ("/stop_memory/%d/%d", mcnt, *p++); break; case duplicate: printf ("/duplicate/%d", *p++); break; case anychar: printf ("/anychar"); break; case charset: case charset_not: { register int c; printf ("/charset%s", (re_opcode_t) *(p - 1) == charset_not ? "_not" : ""); assert (p + *p < pend); for (c = 0; c < *p; c++) { unsigned bit; unsigned char map_byte = p[1 + c]; putchar ('/'); for (bit = 0; bit < BYTEWIDTH; bit++) if (map_byte & (1 << bit)) printchar (c * BYTEWIDTH + bit); } p += 1 + *p; break; } case begline: printf ("/begline"); break; case endline: printf ("/endline"); break; case on_failure_jump: extract_number_and_incr (&mcnt, &p); printf ("/on_failure_jump/0/%d", mcnt); break; case on_failure_keep_string_jump: extract_number_and_incr (&mcnt, &p); printf ("/on_failure_keep_string_jump/0/%d", mcnt); break; case dummy_failure_jump: extract_number_and_incr (&mcnt, &p); printf ("/dummy_failure_jump/0/%d", mcnt); break; case push_dummy_failure: printf ("/push_dummy_failure"); break; case maybe_pop_jump: extract_number_and_incr (&mcnt, &p); printf ("/maybe_pop_jump/0/%d", mcnt); break; case pop_failure_jump: extract_number_and_incr (&mcnt, &p); printf ("/pop_failure_jump/0/%d", mcnt); break; case jump_past_alt: extract_number_and_incr (&mcnt, &p); printf ("/-

```

TIMMY LUTZ

WIE UNTERNEHMEN MOBILE ENDGERÄTE  
ERFOLGREICH IN BESTEHENDE  
IT-INFRASTRUKTUREN INTEGRIEREN

MOBILE DEVICE MANAGEMENT  
IN ZEITEN VON  
MOBILEM ARBEITEN

**Timmy Lutz**

**Mobile Device Management in  
Zeiten von mobilem Arbeiten**

**Wie Unternehmen mobile Endgeräte  
erfolgreich in bestehende  
IT-Infrastrukturen integrieren**

**Bibliografische Information der Deutschen Nationalbibliothek:**

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

**Impressum:**

Copyright © Studylab 2021

Ein Imprint der GRIN Publishing GmbH, München

Druck und Bindung: Books on Demand GmbH, Norderstedt, Germany

Coverbild: GRIN Publishing GmbH | Freepik.com | Flaticon.com | ei8htz

## Management Summary

Immer mehr Mitarbeiter möchten ihre privaten mobilen Endgeräte auch geschäftlich nutzen können. Zudem ist es für ein Unternehmen in der heutigen Wirtschaft notwendig, Innovationen als Erster umzusetzen, um sich Wettbewerbsvorteile zu verschaffen und auf dem Markt führend zu sein. Dem Wunsch der Mitarbeiter, eine simultane Umgebung gegenüber dem Desktoparbeitsplatz mit mobilen Endgeräten im Unternehmen zu ermöglichen, sollte nachgegangen werden. Alle Prozesse und Einrichtungen eines Unternehmens, die dem Zweck dienen, mobile Endgeräte in die Geschäftsprozesse einzubinden, fasst man unter dem Begriff mobile IT-Infrastruktur zusammen.

Um eine wirksame Strategie für ein Unternehmen zu entwickeln, ist es unabdingbar, sich auch Gedanken über Mobile Applikation Management (MAM), Mobile Informationen Management (MIM), Mobile Security Management (MSM) und Mobile Content Management (MCM) zumachen. Mobile Device Management (MDM) ist eine Art Sicherheitssoftware, die von einer IT-Abteilung verwendet wird, um die mobilen Endgeräte von Mitarbeitern zu überwachen, zu verwalten und zu sichern. Beim Mobile Device Management kann vieles aus dem klassischen Device Management adaptiert werden, doch bei einigen Punkten müssen die Konzepte überprüft werden.

Mobile Device Management-Systeme werden hauptsächlich dann eingesetzt, wenn ein Unternehmen seine Daten schützen möchte. In punkto Sicherheit, verliert ein Unternehmen schnell mal die Kontrolle über ihre Daten, da ein Mitarbeiter diese Daten, die sich möglicherweise in seiner privaten Cloud befinden, mit anderen teilen lassen, ohne dass irgendeine Compliance-Anforderung an das mobile Endgerät gestellt wird. Vor allem Grossunternehmen mit hochsensiblen Daten wie Patientendaten oder Daten aus dem Finanzbereich sollten hier strenge Richtlinien erlassen, um die Daten zu schützen. Dabei gilt es jedoch rechtliche Aspekte zu beachten, welche aufgrund der Tatsache, dass sich das mobile Endgerät möglicherweise in privatem Besitz befindet nicht zu verachten sind.

Es bestehen Verschiedenste Ansätze wie man einem Mitarbeiter die Arbeit ausserhalb des Unternehmensnetzwerkes vereinfachen kann. Man spricht hier von Bring your own Device (BYOD) oder Coporate owned personally enabled (COPE). Dabei sind die Ansätze dementsprechend unterschiedlich in deren Anschaffung und Verwaltung der mobilen Endgeräte.

Die dafür verwendeten Mobile Device Management Server stehen entweder im Unternehmen selbst oder werden über einen Cloud Service als SaaS angemietet.

# Inhaltsverzeichnis

<b>Management Summary</b> .....	<b>III</b>
<b>Abkürzungsverzeichnis</b> .....	<b>VIII</b>
<b>1 Einleitung</b> .....	<b>1</b>
<b>2 Problemstellung</b> .....	<b>2</b>
<b>3 Ziele der Arbeit</b> .....	<b>5</b>
<b>4 Grundlagen des Mobile Computing</b> .....	<b>6</b>
4.1 Klassifizierung mobiler Endgeräte .....	6
4.2 Definition .....	8
4.3 Betriebssysteme für mobile Endgeräte .....	9
<b>5 Grundlagen des Mobile Device Management</b> .....	<b>12</b>
5.1 Mobile Device Management (MDM) .....	12
5.2 Mobile Applikation Management (MAM) .....	14
5.3 Mobile Information Management (MIM).....	14
5.4 Mobile Security Management (MSM).....	15
5.5 Mobile Content Management (MCM) .....	15
5.6 Enterprise Mobile Management (EMM).....	16
<b>6 Differenzen zwischen mobilen und herkömmlichen IT-Infrastrukturen</b> .....	<b>18</b>
6.1 Ausrichtung IT.....	18
6.2 Netze und aktive Komponenten .....	19
6.3 Energiemanagement.....	19
6.4 Verwaltung.....	20
6.5 Incident und Problemmanagement .....	21
6.6 Business Continuity und Notfallplanung.....	22
6.7 Audits .....	22

<b>7 MDM und ITIL .....</b>	<b>24</b>
7.1 Servicestrategie .....	25
7.2 Service Design.....	25
7.3 Service Transition.....	25
7.4 Service Operation .....	25
7.5 Kontinuierliche Serviceverbesserung (CSI) .....	26
7.6 Mobile Endgeräte in ITIL.....	26
<b>8 Konzept für Mobile Device Management .....</b>	<b>32</b>
8.1 Bring your own Device .....	32
8.2 Corporate Owned Personally Enabled.....	33
8.3 Take it or leave it.....	34
8.4 Vergleich der Konzepte .....	34
8.5 Architekturen .....	36
8.6 Marktübersicht.....	40
8.7 Unternehmensbefragung zur Marktsituation .....	49
8.8 SWOT-Analyse .....	53
<b>9 IT-Management und Strategie.....</b>	<b>54</b>
9.1 Verteilung der Arbeitslast für die Verwaltung mobiler Endgeräte .....	54
9.2 Mögliche Fehler in der IT-Strategie.....	55
<b>10 Rechtliche Sicht .....</b>	<b>56</b>
10.1 Datenrichtlinie .....	56
10.2 Arbeitsrecht.....	56
10.3 Datenschutz und Compliance .....	58
10.4 Immaterialgüterrecht und Lizenzrechte.....	59
10.5 Haftung.....	60
<b>11 IT-Sicherheit .....</b>	<b>62</b>
11.1 Grundsätzliches .....	63
11.2 Organisatorische Sicherheitsmassnahmen .....	66

11.3 Schwachstellen und Risiken .....	68
<b>12 Wirtschaftliche Erkenntnisse .....</b>	<b>72</b>
12.1 On Premise oder SaaS.....	72
<b>13 Schlussfolgerung .....</b>	<b>80</b>
<b>Literaturverzeichnis.....</b>	<b>82</b>
<b>Abbildungsverzeichnis .....</b>	<b>85</b>
<b>Tabellenverzeichnis.....</b>	<b>86</b>
<b>Anhang .....</b>	<b>87</b>
MDM Fragebogen.....	87
Begründung der Themenwahl .....	96
Bezug zu Unterrichtsfächern .....	96
Fragestellungen .....	96
Absicht der Arbeit .....	97
Ausformulierte Ziele der Arbeit .....	97
Arbeitsvorgehen.....	98
Informationsbeschaffung / Quellen .....	98



## Abkürzungsverzeichnis

Abkürzung	Erklärung
BCM	Business Continuity Management
BIA	Business Impact Analysis
BES	BlackBerry Enterprise Server
BYOD	Bring Your Own Device
CI	Configuration Item
CMDB	Configuration Management Database
CMS	Configuration Management System
COPE	Corporate Owned – Personally Enabled
CRM	Customer Relationship Management
CSI	Continual Service Improvement
CSS	Cascading Style Sheets
EMM	Enterprise Mobility Management
GPS	Global Positioning System
GSM	Global System for Mobile Communications
IDE	Integrated Development Environment
IoT	Internet of Things
ITIL	IT Infrastructure Library
ITSM	IT Service Management
LTE	Long Term Evolution
MAM	Mobile Application Management
MCM	Mobile Content Management
MDM	Mobile Device Management
MIM	Mobile Information Management
MSM	Mobile Security Management
OS	Operation System
OTA	Over the Air

PIM	Personal Information Manager
RAM	Random Access Memory
RIM	Research in Motion
SaaS	Software as a Service
SACM	Service Asset Configuration Management
SDK	Software Development Kit
VPN	Virtual Private Network

