

자동화, 어디부터 어디까지
가능한가?



KOREA DevOps
Meetup '19

Jin Gi KONG
Developer Advocate

IBM **Developer**



Jin Gi KONG

Development

- Python, Javascript, C/C++, Java, PHP, scripts
- Django, Express.js
- PostgreSQL, MySQL, Oracle
- Apache, nginx
- Docker, K8s, Cloud Foundry

Server / Hardware

- Linux, FreeBSD, Windows
- Schematics, PCB artwork, SMD
- MCU: Arduino, Micropython / ESP32, ESP8266, Uno, Pro micro

Network / Security

- VLAN, switching, BGP, OSPF, RIP / Cisco, Juniper
- TCP, UDP, HTTP, DNS, ICMP, Websocket / Wireshark
- X86 ASM, ARM ASM, Z80 ASM / Softice, W32Dasm, IDA
- iptables, ipfw, pf

Entrepreneur

- Startup founder
 - HR
 - accounting
 - LLC
- Corporation in US

자동화를 시작하는 타이밍

같은 작업을
5번 이상
반복해야 할 때

비생산적 일을
생산적으로
바꾸고 싶을 때

다른 사람의
삽질이 너무
안타까울 때

잉여력이
폭발할 때

게임내 동작 반복: Ultima Online (1998)

- MMORPG 의 시작
- 광석 캐기, 마법 스킬 올리기 등의 반복 작업 필요
- 보조 프로그램 등장: UO Assist, UO Extreme
 - 메모리 후킹, detour 등으로 게임 클라이언트 접근
 - 조건에 따라 연속 마법 등 사용 가능

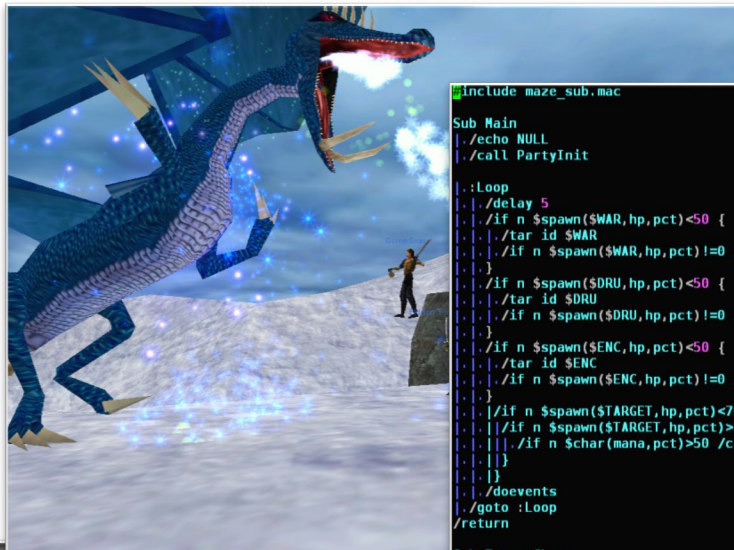


출처: 초록불의 잡학다식, 울티마 온라인 여행기

<http://orumi.egloos.com/3533543>

게임 파티 자동화: Everquest (2002)

- MMORPG, 철저한 클래스 제도
- 어려운 렵업과 100명이 넘는 대규모 레이드
- ShowEQ
 - 패킷 스니핑을 통해 지도 표시
- MacroQuest, Xybotlog, MQPerl
 - 클라이언트메모리 후킹
 - 명령어 강제 실행



```
include maze_sub.mac

Sub Main
  ./echo NULL
  ./call PartyInit

  ./Loop
  ./delay 5
  ./if n $spawn($SWAR, hp, pct) < 50 {
    ./tar id $SWAR
    ./if n $spawn($SWAR, hp, pct) != 0 ./call Cast "Complete Healing"
  }
  ./if n $spawn($SDRU, hp, pct) < 50 {
    ./tar id $SDRU
    ./if n $spawn($SDRU, hp, pct) != 0 ./call Cast "Supernal Remedy"
  }
  ./if n $spawn($SENC, hp, pct) < 50 {
    ./tar id $SENC
    ./if n $spawn($SENC, hp, pct) != 0 ./call Cast "Supernal Remedy"
  }
  ./if n $spawn($TARGET, hp, pct) < 70 {
    ./if n $spawn($TARGET, hp, pct) > 30 {
      ./if n $char(mana, pct) > 50 ./call DD Judgement
    }
  }
  }
  }
  ./doevents
  ./goto :Loop
./return

Sub Event Chat
  ./varset v9 "$p2"

  ./if $p0==group {
    ./if "$v9"=="virtue" {
      ./tar $p1
      ./call Cast "Virtue"
    }
    ./if "$v9"=="gaego" {
      ./call Cast "Blessing of Aegolism"
    }
    ./if "$v9"=="inc" {
      ./call SetTarg
      ./tar id $SWAR
      ./call Cast "Supernal Elixir"
    }
    ./if "$v9"=="dr" {
      ./targ id $SWAR
      ./call Cast "Resist Disease"
    }
    ./if "$v9"=="done" {
      ./if n $spawn($SWAR, hp, pct) < 80 {
        ./tar id $SWAR
        ./if n $spawn($SWAR, hp, pct) > 0 ./call Cast "Supernal Remedy"
      }
      ./delay 50
      ./call BacktoCamp
    }
  }
./return
```

UX 테스트(?) 자동화: Online casino

- Flash game
- PyWin32
- Color average
- 버튼 검색: Pixel neighborhood
- 수정.. 수정..
- pyw 796 줄

```

## blackjack ##
H = 'H'
S = 'S'
R = 'R'
DH = 'DH'
DS = 'DS'
P = 'P'

bjmap = {
  4: [H, H, H, H, H, H, H],
  5: [H, H, H, H, H, H, H],
  6: [H, H, H, H, H, H, H],
  7: [H, H, H, H, H, H, H],
  8: [H, H, H, H, H, H, H],
  9: [H, DH, DH, DH, DH, DH, H],
  10: [DH, DH, DH, DH, DH, DH, D],
  11: [DH, DH, DH, DH, DH, DH, D],
  12: [H, H, R, S, S, S, H],
  13: [S, S, S, S, S, S, H],
  14: [S, S, S, S, S, S, H],
  15: [S, S, S, S, S, S, H],
  16: [S, S, S, S, S, S, H],
  17: [S, S, S, S, S, S, S],
  18: [S, S, S, S, S, S, S],
  19: [S, S, S, S, S, S, S],
  20: [S, S, S, S, S, S, S],
  # A, X
  130: [H, H, H, DH, DH, H, H],
  140: [H, H, H, DH, DH, H, H],
  150: [H, H, DH, DH, DH, H, H],
  160: [H, H, DH, DH, DH, H, H],
  170: [H, DH, DH, DS, DS, DS, S],
  180: [S, DS, DS, DS, DS, S, S],
  190: [S, S, S, S, S, S, S],
  200: [S, S, S, S, S, S, S],
  # Split
  49: [P, P, P, P, P, P, H],
  69: [P, P, P, P, P, P, P],
  89: [H, H, H, P, P, P, H],
  109: [DH, DH, DH, DH, DH, D, D],
  129: [P, P, P, P, P, P, P],
  149: [P, P, P, P, P, P, P],
  169: [P, P, P, P, P, P, P],
  189: [P, P, P, P, P, P, S],
  209: [S, S, S, S, S, S, S],
  29: [P, P, P, P, P, P, P]
}

"casino.pyw" 796  --2%--

```

왜 자동화를 하는가?

시간 단축

생산성 향상

프로세스 개선

스킬업

빌드 스크립팅:

Vim 빌드, Bash 스크립팅

```
:map <F5> :!bash %<CR>
```

```
:map <F5> :!node %<CR>
```

```
:map <F5> :!kubectl apply -f %<CR>
```

```
:map <F5> :!service nginx reload<CR>
```

```
:map <F5> :!~/src/webrepl/webrepl_cli.py -p  
developer % 192.168.0.194:/%:h/<CR>
```

```
:map <F5> :let @l=getline('.')\|exec '!echo  
curl
```

```
http://localhost:8003/members/sms_forwar  
d?m=01012345678 -G --data-urlencode  
"t=@l"
```

```
$ while ;; do screen /dev/tty.SLAB_USBtoUART  
115200 -S usbserial; reset; echo "Screen  
terminated. Waiting for TTY.."; sleep 1; while ;; do  
sleep 0.1; [ -e /tmp/tty.SLAB_USBtoUART.lock ] ||  
break; done; done
```

```
$ while ;; do scp  
10.0.0.11:src/Micropython/firmware/build/chooser  
.bin /Users/arcturus/src/MicroPython/firmware/ &&  
touch /tmp/tty.SLAB_USBtoUART.lock; screen -X  
quit; esptool.py --baud 1500000 --port  
/dev/cu.SLAB_USBtoUART write_flash -z 0x10000  
~/src/MicroPython/firmware/chooser.bin ; rm -f  
/tmp/tty.SLAB_USBtoUART.lock; read; done
```


Vim 스크립팅: Linux 커널 문서 자동화 (2003)

- CC 인증작업
- 구현검증명세서
- Linux 커널에 대해 요구!?!?
- 주석처리용 Vim script 작성
 - 커널 소스에 직접 주석 생성 단축키
 - 소스로부터 직접 PDF 생성
- 퇴사 후 유지보수 불가...

Bash 스크립팅: CDMA Android 롬 수정

- ROM 분류, 롬 압축 해제 및 분리
- boot 이미지 unpack, ramdisk 패치, CLASSPATH 수정, selinux 적용
- recovery 이미지 OTA key 수정
- system 이미지 unpack, smali 추출, framework 수정, repack, dexopt
- cache 이미지 CSC 수정
- MD5 생성
- Bash script 418 줄 + Python 300 줄

```
repack_boot ()
{
    .....IMG="$1"
    .....IMGTYPE="$2"
    .....NEWBOOTIMG="$3"
    .....BOOTDIR=$(IMGTYPE)
    .....RANDISKDIR=$(BOOTDIR)/ramdisk"
    .....BIMG=$(basename $IMG)
    .....log "Repack ${IMGTYPE} image"
    .....cd ${RANDISKDIR}
    .....find . | cpio -o -H newc | gzip > ../${BIMG}-newrand
    .....cd -
    .....DTCMD=""
    .....if [ -f ${BOOTDIR}/${BIMG}-dt ]; then
    .....DTCMD="--dt ${BOOTDIR}/${BIMG}-dt"
    .....fi
    .....${MKBOOTIMG} \
    .....-kernel ${BOOTDIR}/${BIMG}-zImage \
    .....-ramdisk ${BOOTDIR}/${BIMG}-newramdisk.gz
    .....-cmdline "${cat ${BOOTDIR}/${BIMG}-cmdline}
    .....-base 0x$(cat ${BOOTDIR}/${BIMG}-base) \
    .....-pagesize $(cat ${BOOTDIR}/${BIMG}-pagesize)
    .....-ramdisk offset 0x$(cat ${BOOTDIR}/${BIMG}-
    .....tags offset 0x$(cat ${BOOTDIR}/${BIMG}-tag
    .....${DTCMD} \
    .....-o ${NEWBOOTIMG}
    .....mv -f $(IMG) $(MGBACKUPDIR)
    .....mv -f ${NEWBOOTIMG} ${RANDIR}
}

patch_boot ()
{
    .....cat >> ${BOOTDIR}/ramdisk/init.rc << __EOF__
}

service mmsrouted /system/bin/mmsrouted
class main
    socket mmsroute stream 660 root root
    socket aaa stream 660 root system
    user root
    group root
__EOF__
.....log "Extract BOOTCLASSPATH"
.....grep BOOTCLASSPATH ${RANDISKDIR}/${BCPSRC} | head -1
.....cat > ${BCPL1}
.....if [ -f "${RANDISKDIR}/${FILECONTEXTSSRC}" ]; then
.....log "Extract SELinux file contexts"
.....cp ${RANDISKDIR}/${FILECONTEXTSSRC} ${WORKDIR}
.....SELINUXOPT="--S ${WORKDIR}/${FILECONTEXTSSRC}
.....fi

patch_recovery ()
{
    .....cat ${OTAKEYV2} > ${RANDISKDIR}/res/keys
}

modify_boot ()
{
    .....IMG=$1
    .....IMGTYPE=$2
    .....FUNC=$3
    .....unpack_boot ${IMG} ${IMGTYPE}
    .....${FUNC}
    .....NEWBOOTIMG="${WORKDIR}/${basename ${IMG}}"
    .....repack_boot ${IMG} ${IMGTYPE} ${NEWBOOTIMG}
    .....log "Cleanup ${IMGTYPE}"
    .....#rm -fr ${BOOTDIR}
}

modify_img()
{
    .....IMG=$1
```

```
#!/usr/bin/env python
from difflib import unified_diff
import os
from os.path import join, isfile, exists
import re
from shutil import copyfile, copytree
import sys

VER = "1.0.0"
MIN = "4.3"
MAX = "4.4.4"

FILENAME = 0
PATCHES = 1
FILESET = 2
CONDITION = 3

PATCHFILE = 'arcysystem.patch'

patchset = [
    # 4.4
    [
        'com/android/internal/telephony/ImsSMSDispatcher.smali', # filename
        [ # patches
            ( # search / replacement
                r'/CdmaInboundSmsHandler;{(["\\"])}',
                r'/ArcyCdmaInboundSmsHandler;\\',
            ),
        ],
        [ # fileset
            'com/android/internal/telephony/cdma/ArcyCdmaInboundSmsHandler.smali',
            'com/android/internal/telephony/cdma/ArcyCdmaSms.smali',
        ],
        ('com/android/internal/telephony/ImsSMSDispatcher.smali', True),
    ],
    [
        'com/android/internal/telephony/cdma/CdmaInboundSmsHandler.smali',
        [
            (
                '.method private constructor <init>',
                '.method constructor <init>',
            ),
        ],
        [
            ('com/android/internal/telephony/ImsSMSDispatcher.smali', True),
        ],
    ],
    # 4.3
    [
        'com/android/internal/telephony/cdma/CDMAPhone.smali',
        [
            (
                'Lcom/android/internal/telephony/cdma/CdmaSMSDispatcher;',
                'Lcom/android/internal/telephony/cdma/ArcyCdmaSMSDispatcher;',
            ),
            (
                'Lcom/android/internal/telephony/cdma/ArcyCdmaSms;',
                'Lcom/android/internal/telephony/cdma/ArcyCdmaSms;',
            ),
        ],
        [
            'com/android/internal/telephony/cdma/ArcyCdmaSMSDispatcher.smali',
            'com/android/internal/telephony/cdma/ArcyCdmaSms.smali',
        ],
        ('com/android/internal/telephony/ImsSMSDispatcher.smali', False),
    ],
    [
        'com/android/internal/telephony/cdma/CdmaSMSDispatcher.smali',
        [
            (
                r'".class(.*) final ',
                r'".class\\1 ',
            ),
        ],
    ],
]
```

자동화 매니아의 특징

게으르다.

버그는 용납하나
실수는 용납하지
못한다.

몸보다 머리가
힘든 것이 낫다.

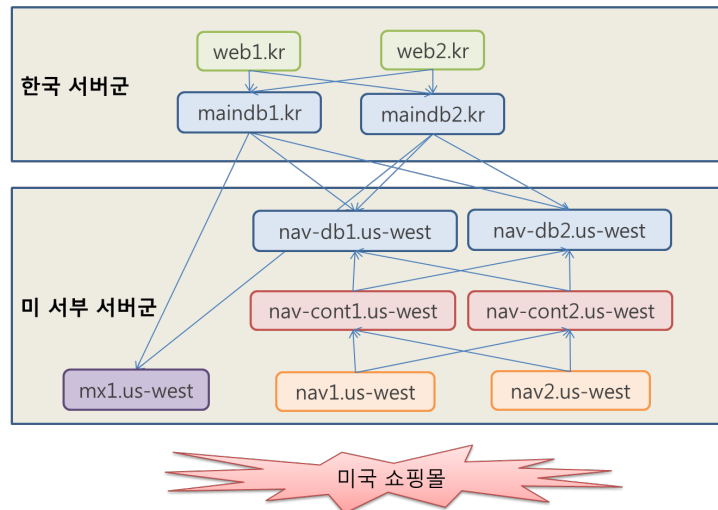
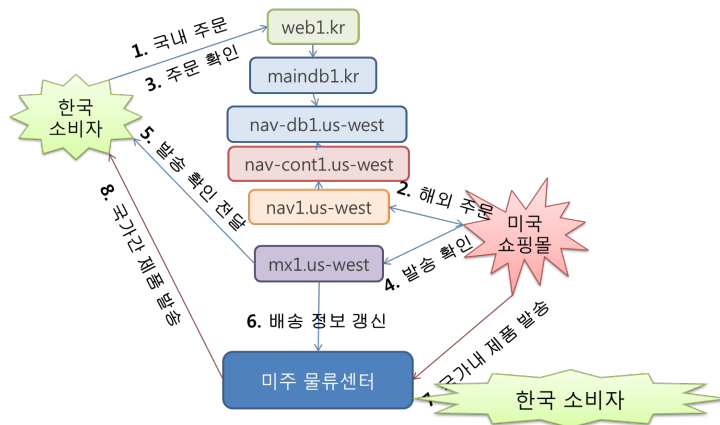
비효율을 참지
못한다.

온라인 구매 자동화: shopst.kr (2012)

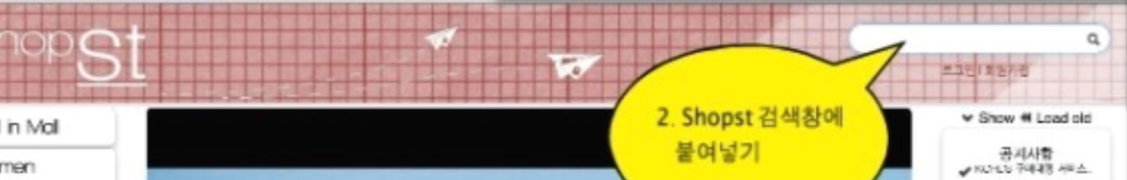
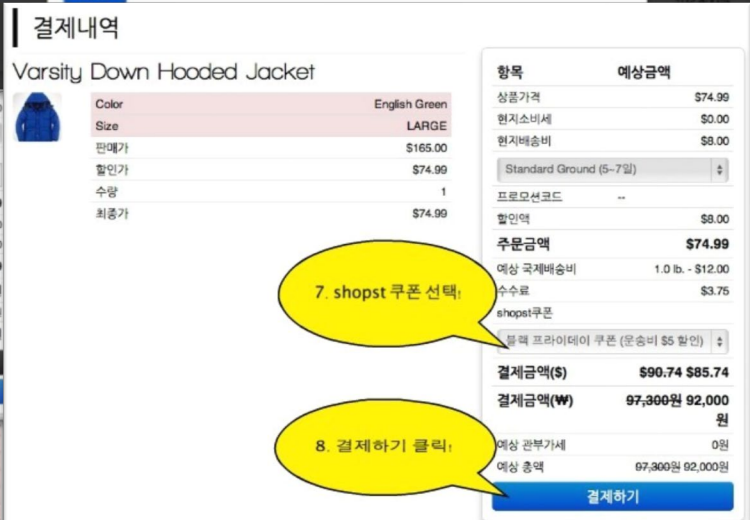
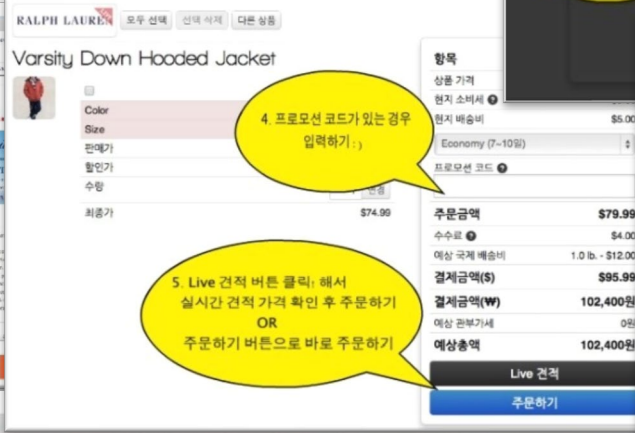
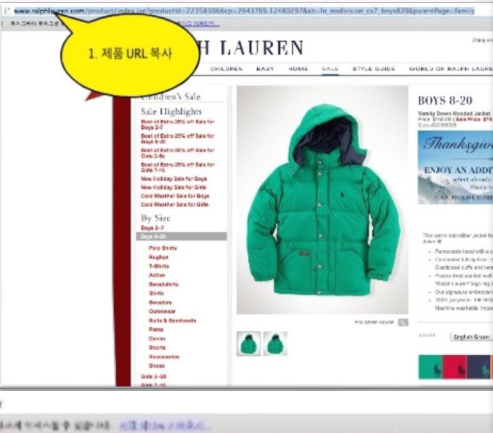
- 해외 직구로 배송대행 및 구매대행 유형
- 자동화로 구매대행을 편하게 할 수 있지 않을까?

• ~~시스템이 일하고 운영자는 노는 시스템~~

- 해외몰 크롤링을 통한 메타디비
- State machine 기반 코딩 없는 구매 프로세스
- 구매 및 결제 자동화



shopst.kr: 구매 프로세스



shopst.kr: 뒤에서는..



자동화 유지보수의 어려움

대상의 변화

인력의 변동

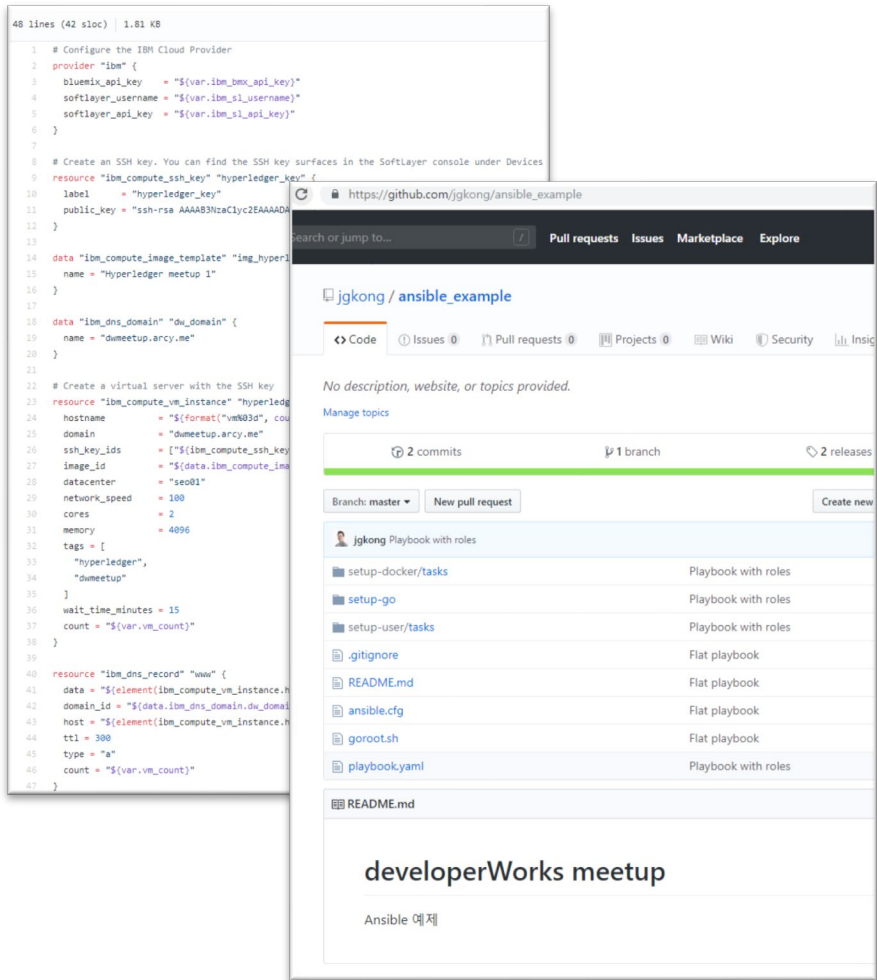
복잡도 증가

이력 유실

인프라 자동화 (IaaC): Terraform & Ansible

- Hyperledger 실습용 VM 50개
- Docker, tty.js, Eclipse Orion Editor
- Terraform 으로 대량 생성
- Ansible 로 template 생성
- node.js 스크립트로 접속키, 비밀번호 생성 및 변경
- Meetup 녹화영상

- <https://www.facebook.com/groups/developerWorksKRUG/permalink/2040426779503489/>
- <https://www.facebook.com/groups/developerWorksKRUG/permalink/2040521826160651/>



하드웨어 빌드 자동화: Firmware autoflash

- ESP8266 80개 플래싱
 - Raspberry PI 사용
 - udev 로 USB 모니터링
 - Admin 서버에 MAC address 기록
 - 초기 비밀번호는 MAC 으로부터 hash 생성
 - 라벨프린터로 MAC, 비밀번호 출력하여 ESP8266 마킹
- ESP32 배지 500개 플래싱
 - Raspberry PI 4대 사용
 - 플래싱 대시보드



시스템간 연동 : 네이버 카페 자동 등업

- 전화번호 인증으로 등업
- SMS, ARS 수신은 착신전환 구별 불가
- 발신인 조작이 불가능한 SMS 발신으로 인증
- 수신 SMS 를 서버로 전달하는 안드로이드 앱
- 네이버 카페 등업 페이지 분석 및 구현
 - Python mechanize 이용
 - 로그인, 등업, 어뷰즈체크
 - 게시물, 댓글: 익명게시판 구현

- 구현상 문제: 부정합한 데이터, TLS1.2

```
path += memberid
url = 'http://{}/{}/'.format(host, path)
#print url
#with requests.session() as s:
#    #r = s.get(url, headers=self.defheaders)
#    #print r.content.decode('cp949')
res = self.get(url)
match = re.search(r'levelicon/./_(.*)\.gif', res.get_data())
lv = 0
if match:
    lv = int(match.group(1))
return lv

def do_levelup(self, memberid, level):
    host = '...'
    path = '/...?m=view&fromUpService=member&model.clubid=12345678&model.memberid='
    path += memberid
    url = 'http://{}/{}/'.format(host, path)
    #print url
    res = self.get(url)

    self.br.select_form(nr=0)
    self.br.form['model.memberLevel'] = [str(level)]
    res2 = self.br.submit()

def do_logincheck(self):
    host = '...'
    path = '/...?m=check'
    url = 'http://{}/{}/'.format(host, path)
    #print url
    res = self.get(url)
    data = res.get_data()
    if (data != 'LOGIN'):
        raise MemberException('LoginCheck error: {}'.format(data), 'L')

def do_abusecheck(self, abuseType):
    host = '...'
    path = '/...'
    url = 'http://{}/{}/'.format(host, path)
    data = 'abuseType={}'.format(abuseType)
    res = self.get(url, data)
    data = res.get_data()

    try:
        import json
        json_data = json.loads(data)
        if json_data['isSuccess'] != True or json_data['result']['abuser'] != False:
            raise MemberException('Abuse result is bad', 'A')
    except Exception as e:
        print e
        raise MemberException('Abuse check error: {}'.format(data), 'A')

def do_readarticle(self, articleid):
    host = '...'
    path = '/...?clubid=12345678&page=1&boardtype=L&articleid={}&referrerAllArticles=true'.format(articleid)
    url = 'http://{}/{}/'.format(host, path)
    res = self.get(url)
    return res

def do_postcomment(self, data):
    import urllib
```

왜 자동화를 하는가?

시간 단축

생산성 향상

프로세스 개선

스킬업

자동화 언어와 도구들

– Bash

- `$()` `$[]` `${}` `${#}` `${%}` `${//}`
- If, while, `&&`, `||`, `;`,
- awk, sed, cut, xargs, head/tail, popd/pushd

– Python

– Node.js

– Regular Expression

– PHP, Perl

– RPA

– IaaS

- Terraform

- Ansible

– CI/CD

- Jenkins, Travis

- Selenium

- PhantomJS

– Kubernetes

Thank you.

Jin Gi KONG

Developer Advocate

—

jgkong@kr.ibm.com

