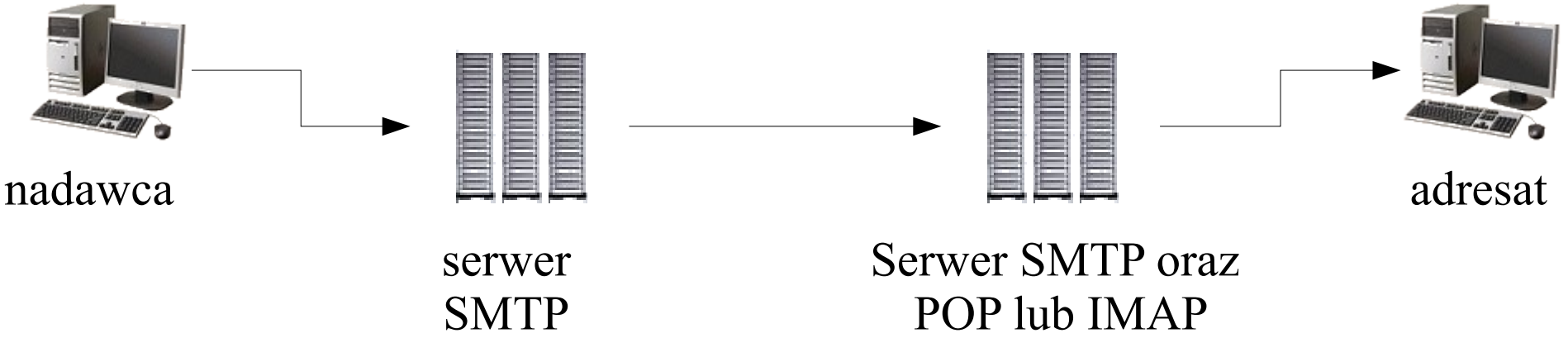
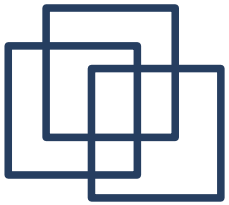
Plan wykładu

1. Poczta elektroniczna
- protokół SMTP,
- protokół POP,
- protokół IMAP.
2. Zdalna praca - Telnet.
3. Transfer plików – usługa FTP.

Poczta elektroniczna

1. Wysyłanie wiadomości e-mail – protokół SMTP (*Simple Mail Transfer Protocol*).
2. Odbiór wiadomości e-mail – propokół POP (*Post Office Protocol*) lub IMAP (*Internet Mail Access Protocol*).





Protokół SMTP

Protokół SMTP [RFC 2821, 821, 2554] jest wykorzystywany do wysyłania wiadomości e-mail. Port 25.

Przykładowe połączenie:

```
[serwer] 220 theta.uoks.uj.edu.pl ESMTP Sendmail
         8.13.1/8.13.1; Mon, 11 Oct 2004 10:19:30 +0200
         (CEST)
[klient] EHLO [127.0.0.1]<CRLF>
[serwer] 250-theta.uoks.uj.edu.pl Hello rhamnus.if.uj.edu.pl
         [149.156.74.158], pleased to meet you
         250-ENHANCEDSTATUSCODES
         ...
         250-AUTH LOGIN PLAIN
         ...
         250 HELP
[klient] AUTH PLAIN <BASE64(login<0x00>has  o)><CRLF>
```

Kodowanie Base64

Alfabet Base64 [RFC 1521]

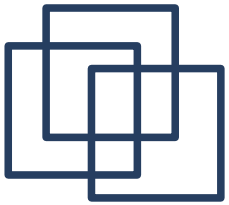
0 A	8 I	16 Q	24 Y	32 g	40 o	48 w	56 4
1 B	9 J	17 R	25 Z	33 h	41 p	49 x	57 5
2 C	10 K	18 S	26 a	34 i	42 q	50 y	58 6
3 D	11 L	19 T	27 b	35 j	43 r	51 z	59 7
4 E	12 M	20 U	28 c	36 k	44 s	52 0	60 8
5 F	13 N	21 V	29 d	37 l	45 t	53 1	61 9
6 G	14 O	22 W	30 e	38 m	46 u	54 2	62 +
7 H	15 P	23 X	31 f	39 n	47 v	55 3	63 /

(pad) =

kodowanie: ala ma kota

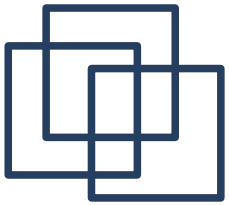
```
a (0x61) l (0x6C) a (0x61)   (0x20) m (0x6D) a (0x61)
01100001 01101100 01100001 00100000 01101101 011000001
  (0x20) k (0x6B) o (0x6F) t (0x74) a (0x61)
00100000 01101011 01101111 01110100 01100001 00
```

wynik: `YWxhIG1hIGtvdGE=`



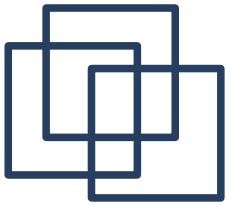
Protokół SMTP

```
[serwer]235 2.0.0 OK Authenticated
[klient]MAIL FROM:<nadawca@adres.email><CRLF>
[serwer]250 2.1.0 <nadawca@adres.email>... Sender ok
[klient]RCPT TO:<adresat@adres.email><CRLF>
[serwer]250 2.1.5 <adresat@adres.email>... Recipient ok
[klient]DATA<CRLF>
[serwer]354 Enter mail, end with "." on a line by itself
```



Protokół SMTP

```
[klient]Date: Mon, 11 Oct 2004 10:19:24 +0200<CRLF>
      User-Agent: cokolwiek<CRLF>
      X-Accept-Language: en-us, en<CRLF>
      MIME-Version: 1.0<CRLF>
      From: ktokolwiek <dowolny@adres.email><CRLF>
      To:  inny@adres.email<CRLF>
      Subject: jakis temat<CRLF>
      Content-Type: text/plain; charset=us-ascii<CRLF>
      Content-Transfer-Encoding: 7bit<CRLF><CRLF>
      tekst testowego maila<CRLF>
      .<CRLF>
[serwer]250 2.0.0 Message accepted for delivery
[klient]QUIT<CRLF>
[serwer]221 2.0.0 theta.uoks.uj.edu.pl closing connection
[serwer]    Zakończenie połączenia.
```



Protokół POP

Protokół POP [RFC 1939] jest używany do odbierania wiadomości e-mail znajdujących sie na serwerze. Port 110.

Przykładowe połączenie:

[serwer] +OK <processID.clock@hostname><CRLF>

[klient] USER u ytkownik<CRLF>

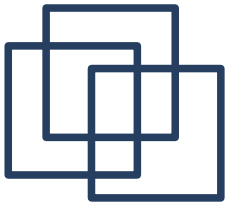
[serwer] +OK<CRLF>

[klient] PASS has e<CRLF>

[serwer] +OK<CRLF>

[klient] STAT<CRLF>

[serwer] 2 540<CRLF> // ilość wiadomości i ich rozmiar w oktetach



Protokół POP

[klient]LIST<CRLF>

[serwer]+OK<CRLF>1 120<CRLF>2 420<CRLF>

[klient]LIST 3<CRLF>

[serwer]+ERR ewentualnie powód b d<CRLF>

[klient]RETR 1<CRLF>

[serwer]+OK<CRLF>
<wiadomo e-mai<CRLF>

[klient]DELE 1<CRLF>

[serwer]+OK<CRLF>

[klient]QUIT<CRLF>

[serwer]+OK<CRLF>

[serwer]Zako czenie po czenia

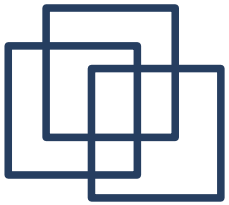
Protokół POP

Inne komendy:

- RSET – odznacza wszystkie wiadomości przeznaczone do skasowania,
- NOOP – podtrzymuje połączenie,

Komendy opcjonalne:

- TOP msg n – zwraca nagłówek oraz n początkowych lini wiadomości msg,
- UIDL msg – zwraca unikalny identyfikator wiadomości,
- APOP login kod – inny sposób autoryzacji. Pierwszy argument stanowi nazwę użytkownika, drugi to kod MD5 z połączenia <processID.clock@hostname> oraz ustalonego wcześniej wyrażenia znanego klientowi i serwerowi.



Kod MD5

MD5 (Message Digest Algorithm) [RFC 1321] oblicza 128 bitowy kod dla zbioru danych. Kod ten może być traktowany jak cyfrowy podpis (fingerprint) danych. Na podstawie kodu praktycznie nie można odtworzyć oryginalnych danych ponieważ wygenerowanie informacji posiadającej określony kod MD5 wymaga w praktyce 2^128 operacji.

Opis algorytmu:

dane wejściowe: **`m[0] m[1] m[2] ... m[b-2] m[b-1]`** – b bitów

1. **Uzupełnianie danych.** Dopisujemy na końcu bity „100000....” tak aby długość danych modulo 512 była równa 448. Uzupełnienie wykonujemy także wtedy, gdy dane wejściowe mają już taką długość.

Kod MD5

2. **Podział na 32 bitowe słowa.** Dopisujemy początkowy rozmiar danych (b) zapisany w dwóch 32-bitowych słowach, mniej znaczące słowo najpierw (1TB = 243 bitów). Niech **`M[0] M[1] ... M[N-1]`** oznaczają kolejne 32 bitowe słowa otrzymanych danych. N jest wielokrotnością 16.
3. **Inicjalizacja obliczeń.** Niech:

```
A = 01 23 45 67
B = 89 ab cd ef
C = fe dc ba 98
D = 76 54 32 10
F(X,Y,Z) = X and Y or not(X) and Z
G(X,Y,Z) = X and Z or Y and not(Z)
H(X,Y,Z) = X xor Y xor Z
I(X,Y,Z) = Y xor (X v not(Z))
T[i] = 4294967296 abs( sin(i) ) //  0 < i < 65;
```

Kod MD5

4. Obliczenia.

```
For i = 0 to N/16-1 do
   For j = 0 to 15 do
      Set X[j] to M[i*16+j].
   end /* p  tla po j */
   AA = A; BB = B; CC = C; DD = D

   /* Cz       1. */
   /* Niech [abcd k s i] oznacza operacj
      a = b + ((a + F(b,c,d) + X[k] + T[i]) <<< s). Wykonaj:*/
   [ABCD  0  7  1] [DABC  1 12  2] [CDAB  2 17  3] [BCDA  3 22  4]
   [ABCD  4  7  5] [DABC  5 12  6] [CDAB  6 17  7] [BCDA  7 22  8]
   [ABCD  8  7  9] [DABC  9 12 10] [CDAB 10 17 11] [BCDA 11 22 12]
   [ABCD 12  7 13] [DABC 13 12 14] [CDAB 14 17 15] [BCDA 15 22 16]

   /* Cz       2. */ ... /* Cz       3. */ ... /* Cz       4. */

   A = A + AA; B = B + BB; C = C + CC; D = D + DD
end /* p  tla po i */
```

Kod MD5

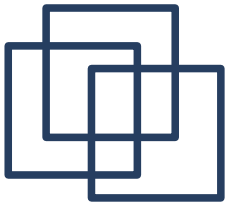
4. Wynik. Kodem MD5 jest wyrażenie A B C D zapisane w kolejności od najmniej znaczącego bitu A do najbardziej znaczącego bitu D.
W systemie linux istnieje komenda „md5” wyliczająca kod dla podanego pliku.

Protokół IMAP

Protokół IMAP [RFC 2060] jest używany do przeglądania wiadomości e-mail przechowywanych na serwerze. Port 143.

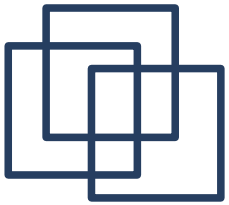
Przykładowe połączenie:

```
[serwer] * OK IMAP4rev1<CRLF>
[klient] 1 capability
[serwer] * CAPABILITY IMAP4REV1 AUTH=LOGIN<CRLF>
      1 OK CAPABILITY completed<CRLF>
[klient] 2 authenticate login<CRLF>
[serwer] + VXNlciBOYW1lAA==<CRLF>   // Base64(User Name)
[klient] <Base64(u  ytkownik)=<CRLF>
[serwer] UGFzc3dvcmQA<CRLF>         // Base64(Password)
[klient] <Base64(has  o)=<CRLF>
```



Protokół IMAP

```
[serwer]* OK [ALERT] Account expires in 14 day(s)<CRLF>
inna autoryzacja: 2 login ”u  ytkownik” ”has  e”<CRLF>
[klient]3 select ”INBOX”<CRLF>
[serwer]* FLAGS (\Draft \Answered \Flagged \Deleted \Seen
        \Recent)
        * 2 EXISTS
        * 0 RECENT
        * OK [UIDVALIDITY 1053533958] Ok
        3 OK [READ-WRITE] Ok<CRLF>
[klient]4 UID FETCH 1:* (FLAGS)<CRLF>
[serwer]* 1 FETCH (UID 1082 FLAGS (\Seen))<CRLF>
        * 2 FETCH (UID 1210 FLAGS (\Answered \Seen))<CRLF>
        4 OK FETCH completed.<CRLF>
```



Protokół IMAP

```
[klient]5 UID FETCH 1210 (UID RFC822.SIZE BODY[])<CRLF>
[serwer]* 283 FETCH (UID 1583 RFC822.SIZE 1180 BODY[]
        {1180}<CRLF>
[serwer]<wiadomo     email><CRLF>
        * 283 FETCH (FLAGS (\Seen \Recent))<CRLF>
        5 OK FETCH completed.<CRLF>


[klient]6 LOGOUT<CRLF>
[serwer]* BYE<CRLF>
        6 OK logout completed.<CRLF>
[serwer]Zako  czenie po    czenia
```

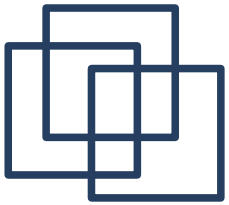
Protokół IMAP

Wiadomości na serwerze IMAP są grupowane w katalogach, podobnie jak pliki na dysku. Większość komend protokołu IMAP umożliwia poruszanie się i wykonywanie odpowiednich operacji na tej strukturze:

- **CREATE**, **DELETE**, **RENAME** – operacje na katalogach (folderach),
- **STORE**, **COPY**, **EXPUNGE** – operacje na wiadomościach.
- **X...** - komendy rozszerzające funkcje protokołu IMAP.

Odpowiedzi serwera:

- **OK** – operacja wykonana pomyślnie,
- **NO** – wystąpił problem,
- **BAD** – niewłaściwa komenda.



Popularne serwery pocztowe

SMTP (Mail Transfer Agent):

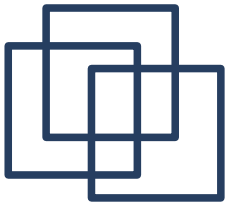
- Sendmail (http://www.sendmail.org/),
- Qmail (http://www.qmail.org/),
- Postfix (http://www.postfix.org/),

POP, IMAP

- courier-imap (http://www.courier-mta.org/imap/)

Zarządzanie domenami wirtualnymi

- vpopmail (http://www.inter7.com/index.php?page=vpopmail),
- mysql (http://www.mysql.com).

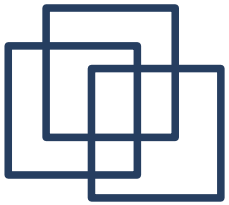


Polecenie Telnet

Protokół Telnet [RFC 854] jest/był wykorzystywany głównie pracy terminalowej poprzez port 23. Za pomocą programu klienta usługi Telnet można się także połączyć z dowolnym innym portem na serwerze, np.

```
telnet theta.uoks.uj.edu.pl 25
```

Komendy wpisywane ze standardowego wejścia (klawiatury) są wysyłane bezpośrednio do wskazanej usługi (SMTP) na serwerze zdalnym. Odpowiedzi są kierowane na standardowe wyjście (ekran).



Protokół FTP

Protokół FTP *(File Transfer Protocol)* [RFC 959] umożliwia przesyłanie plików tekstowych i binarnych. Serwer FTP działa na porcie 21.

[serwer] **220 serwer.adres FTP server ready.**

[klient] **USER użytkownik<CRLF>**

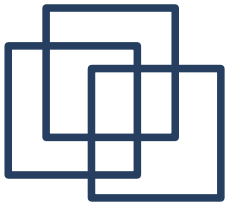
[serwer] **331 Password required for użytkowinik.**

[klient] **PASS hasło<CRLF>**

[serwer] **230 User użytkownik logged in.**

[klient] **SYST<CRLF>**

[serwer] **215 UNIX Type: L8 Version: Compaq Tru64 UNIX V5.0**



Protokół FTP

[klient] **PASV<CRLF>**

[serwer] **227 Entering Passive Mode**

(ip1,ip2,ip3,ip4,port1,port2)

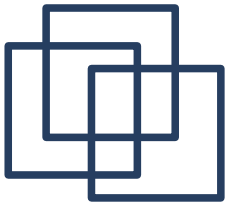
[klient] **LIST<CRLF>**

[serwer] **150 Opening ASCII mode for data connection for /bin/ls (0.0.0.0,0)**

dane są przesyłane na innym porcie. Port 21 będzie przeznaczony tylko dla komunikatów kontrolnych. W celu odebrania danych należy otworzyć nowe połączenie z adresem:

ip1.ip2.ip3.ip4:port1*256+port2.

[serwer] **226 Transfer complete.**



Tryb pasywny i tryb aktywny

Protokół FTP wykorzystuje dwa połączenia TCP. Jedno służy do przesyłania poleceń, drugie do przesyłania danych.

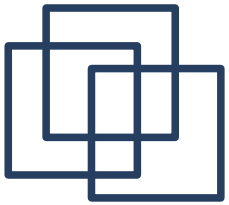
Tryb aktywny:

- port 21 – polecenia,
- port 20 – dane (połączenie z klientem inicjowane przez serwer!).

Tryb pasywny:

- port 21 – polecenia
- port o numerze > 1024 – dane

obydwa połączenia inicjowane przez klienta



Protokół FTP

[klient] **PASV<CRLF>**

[serwer] **227 Entering Passive Mode**

(ip1,ip2,ip3,ip4,port1,port2)

[klient] **RETR jakis.plik<CRLF>**

[serwer] **150 Opening BINARY mode data connection for jakis.plik (0.0.0.0,0)**

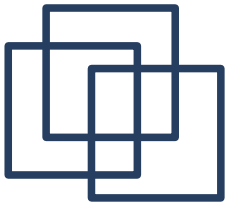
...

[serwer] **226 Transfer complete.**

[klient] **QUIT<CRLF>**

[serwer] **Goodbye.**

[serwer] Zakończenie połączenia.



Serwery FTP

proftpd (http://www.proftpd.org)

vsftpd (http://vsftpd.beasts.org)

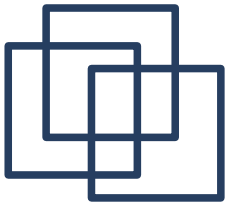
Serwer ftp może korzystać z lokalnej bazy użytkowników (konta na serwerze) lub też autoryzować użytkownika poprzez inny niezależny mechanizm.

Dostęp do ftp bez powłoki: modyfikacja **/etc/passwd**:

```
ciesla:x:1002:21::/var/ftp/ciesla:/bin/false
```

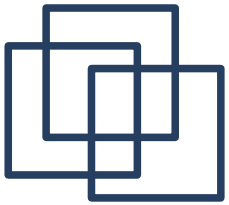
Autoryzacja „lokalna” przez PAM, plik **/etc/pam.d/ftp**

```
#%PAM-1.0
auth       required     /lib/security/pam_listfile.so item=user
  sense=deny file=/etc/ftpusers onerr=succeed
auth       required     /lib/security/pam_unix.so shadow nullok
#auth       required     /lib/security/pam_shells.so
account    required     /lib/security/pam_unix.so
session    required     /lib/security/pam_unix.so.
```



Podsumowanie

Na wykładzie zostały omówione usługi wykorzystywane do odbierania poczty elektronicznej, zdalnej pracy terminalowej oraz transmisji plików tekstowych i binarnych.



Dodatek: Kod MD5

```
/* Część 2. */
    /* Niech [abcd k s i] oznacza operację
         a = b + ((a + G(b,c,d) + X[k] + T[i]) <<< s). */
    [ABCD  1  5 17] [DABC  6  9 18] [CDAB 11 14 19] [BCDA  0 20 20]
    [ABCD  5  5 21] [DABC 10  9 22] [CDAB 15 14 23] [BCDA  4 20 24]
    [ABCD  9  5 25] [DABC 14  9 26] [CDAB  3 14 27] [BCDA  8 20 28]
    [ABCD 13  5 29] [DABC  2  9 30] [CDAB  7 14 31] [BCDA 12 20 32]
    /* Część 3. */
    /* Niech [abcd k s i] oznacza operację
         a = b + ((a + H(b,c,d) + X[k] + T[i]) <<< s). */
    [ABCD  5  4 33] [DABC  8 11 34] [CDAB 11 16 35] [BCDA 14 23 36]
    [ABCD  1  4 37] [DABC  4 11 38] [CDAB  7 16 39] [BCDA 10 23 40]
    [ABCD 13  4 41] [DABC  0 11 42] [CDAB  3 16 43] [BCDA  6 23 44]
    [ABCD  9  4 45] [DABC 12 11 46] [CDAB 15 16 47] [BCDA  2 23 48]
    /* Część 4. */
    /* Niech [abcd k s i] oznacza operację
         a = b + ((a + I(b,c,d) + X[k] + T[i]) <<< s). */
    [ABCD  0  6 49] [DABC  7 10 50] [CDAB 14 15 51] [BCDA  5 21 52]
    [ABCD 12  6 53] [DABC  3 10 54] [CDAB 10 15 55] [BCDA  1 21 56]
    [ABCD  8  6 57] [DABC 15 10 58] [CDAB  6 15 59] [BCDA 13 21 60]
    [ABCD  4  6 61] [DABC 11 10 62] [CDAB  2 15 63] [BCDA  9 21 64]
```